



Universidade do Minho

Escola de Engenharia

Ana Rita Vieira Rodrigues

Retinal Image Quality Assessment using Deep Convolutional Neural Networks

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Informática Médica

Trabalho efetuado sobre a orientação de:

Victor Manuel Rodrigues Alves

Manuel João Ferreira

DECLARAÇÃO

Nome: Ana Rita Vieira Rodrigues

Endereço eletrónico: anaritavr@gmail.com

Cartão de Cidadão: 14912766

Título da dissertação: *Retinal Image Quality Assessment using Deep Convolutional Neural Networks*

Orientadores: Victor Manuel Rodrigues Alves, Manuel João Ferreira

Ano de Conclusão: 2018

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica

Área de Especialização: Informática Médica

Escola de Engenharia

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO

Universidade do Minho, ____/____/____

Assinatura: _____

ACKNOWLEDGMENTS

At the culmination of these years of study, I would like to thank those who accompanied me at all stages, whether they were good or bad.

A special thanks to Professor Victor Alves who guided me, helped throughout the course of this work and always gave me the best advice and points of view. I thank Professor Manuel Ferreira who advised me and made me doubt about several subjects of the dissertation, as well as all the help in the integration of the Medical Imaging team at Neadvance.

I am very grateful to Diana, Nélon, Eva, Pedro Morgado, Pedro Silva, Darya, Ana and all the colleagues of the Industrial Image, since they were fundamental in my journey, always having an advice, a suggestion, another vision, and experience in the most varied subjects and areas.

To my family, parents and my brother, I thank you for giving me support, love, strength, motivation and for having been in all phases of my life. To my uncles, cousins , and grandmother for having been my friends, encouraging me and having always wished me the greatest luck and success.

To my friends of Basic Education, Lila and Tiago, thank you very much for their company, the friendship that has been very fundamental and “borgas” that make me smile so much. To my secondary friends, Mariana and Cláudio, who are still at my side, being real friends.

To my friend Ana and André, who are the most humble and sincere friends I have. To my childhood friends Sara and Francisco, my best friend Juliana, Rita Sousa, João, Paula, Daniel's parents and all the other friends who were present this past year, a warm thank you.

To my university friends, to the great Márcia, Juan, Bruno, Graxas, Rui and Luis, I have a great thank you to give because they have always been very close to me, to study, to make me smile and thrill with my success. From them, I take a friendship for life.

Lastly, I thank my boyfriend Daniel for his patience, for his love, for his motivation, and he has always given me the greatest moral support.

“ Machine Intelligence is the last invention
that Humanity will ever need to make. ”

Nick Bostrom

ABSTRACT

Diabetic Retinopathy (DR) and diabetic macular edema (DME) are the damages caused to the retina and are complications that can affect the diabetic population. Diabetic retinopathy (DR), is the most common disease due to the presence of exudates and has three levels of severity, such as mild, moderate and severe, depending on the exudates distribution in the retina. For screening of diabetic retinopathy or a population-based clinical study, a large number of digital fundus images are captured and to be possible to recognize the signs of DR and DME, it is necessary that the images have quality, because low-quality images may force the patient to return for a second examination, wasting time and possibly delaying treatment.

These images are evaluated by trained human experts, which can be a time-consuming and expensive task due to the number of images that need to be examined. Therefore, this is a field that would be hugely benefited with the development of an automated eye fundus quality assessment and analysis systems. It can potentially facilitate health care in remote regions and in developing countries where reading skills are scarce.

Deep Learning is a kind of Machine Learning method that involves learning multi-level representations that begin with raw data entry and gradually moves to more abstract levels through non-linear transformations. With enough training data and sufficiently deep architectures, neural networks, such as Convolutional Neural Networks (CNN), can learn very complex functions and discover complex structures in the data.

Thus, Deep Learning emerges as a powerful tool for medical image analysis and evaluation of retinal image quality using computer-aided diagnosis.

Therefore, the aim of this study is to automatically assess all the three quality parameters alone (focus, illumination and color), and then an overall quality of fundus images assessment, classifying the images into the classes “accept” or “reject with a Deep Learning approach using convolutional neural networks (CNN). For the overall classification, the following results were obtained: test accuracy=97.89%, SN=97.9%, AUC=0.98 and F_1 -score=97.91%.

RESUMO

A retinopatia diabética (RD) e o edema macular diabético (EMD) são patologias da retina e são uma complicação que pode afetar a população diabética. A retinopatia diabética é a doença mais comum devido à presença de exsudatos e possui três níveis de gravidade, como leve, moderado e grave, dependendo da distribuição dos exsudatos na retina.

Para triagem da retinopatia diabética ou estudo clínico de base populacional, um grande número de imagens digitais de fundo do olho são capturadas e para ser possível reconhecer os sinais da RD e EMD, é necessário que as imagens tenham qualidade, pois imagens de baixa qualidade podem forçar o paciente a retornar para um segundo exame, perdendo tempo e, possivelmente, retardando o tratamento. Essas imagens são avaliadas por especialistas humanos treinados, o que pode ser uma tarefa demorada e cara devido ao número de imagens que precisam de ser examinadas. Portanto, este é um campo que seria enormemente beneficiado com o desenvolvimento de sistemas automatizados de avaliação e análise da qualidade da imagem do fundo de olho. Pode potencialmente facilitar a assistência médica em regiões remotas e em países em desenvolvimento, onde as habilidades de leitura são escassas.

Deep Learning é um tipo de método de *Machine Learning* que envolve a aprendizagem de representações em vários níveis que começam com a entrada de dados brutos e gradualmente se transformam para níveis mais abstratos através de transformações não lineares, para se obterem as previsões. Com dados de treino suficientes e arquiteturas suficientemente profundas, as redes neurais, como as *Convolutional Neural Networks* (CNN), podem aprender funções muito complexas e descobrir estruturas complexas nos dados. Assim, o *Deep Learning* surge como uma ferramenta poderosa para analisar imagens médicas para avaliação da qualidade da retina, usando diagnóstico auxiliado por computador a partir do fundo do olho.

Portanto, o objetivo deste estudo é avaliar automaticamente a qualidade geral das imagens do fundo, classificando as imagens em “aceites” ou “rejeitadas”, com base em três parâmetros principais, como o foco, a iluminação e cor com abordagem de *Deep Learning* usando *convolutional neural networks* (CNN).

Para a classificação geral da qualidade das imagens, obtiveram-se os seguintes resultados: acurácia do teste = 97,89%, SN = 97,9%, AUC = 0,98 e F_1 -score=97.91%.

TABLE OF CONTENTS

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Motivation..... | 3 |
| 1.2 | Objectives and Research Questions..... | 3 |
| 1.3 | Research Methodology | 4 |
| 1.4 | Dissertation Structure | 5 |
| 2 | Medical Background | 7 |
| 2.1 | Eye Anatomy | 9 |
| 2.2 | Retinal Imaging Modalities | 11 |
| 2.1.1 | Digital Fundus Photography..... | 12 |
| 2.1.2 | Fluorescein Angiography..... | 12 |
| 2.1.3 | Optical Coherence Tomography | 13 |
| 3 | Retinal Image Quality Assessment..... | 15 |
| 3.1 | Retinal Image Quality Parameters..... | 17 |
| 3.1.1 | Field Definition | 18 |
| 3.1.2 | Clarity And Focus | 19 |
| 3.1.3 | Visibility of the Macula | 20 |
| 3.1.4 | Visibility of the Optic Disc | 20 |
| 3.1.5 | Artifacts | 20 |
| 3.2 | Retinal Image Quality Assessment State-of-the-Art..... | 21 |
| 3.2.1 | Generic IQA Approaches..... | 22 |
| 3.2.2 | Deep Learning IQA Approaches | 28 |
| 4 | Deep Neural Networks Principles and Fundamentals | 35 |
| 4.1 | Machine Learning and Deep Learning | 37 |
| 4.1.1 | Neural Network | 40 |
| 4.1.2 | Loss Functions..... | 41 |

| | | |
|-------|--|-----|
| 4.1.3 | Activation Functions | 42 |
| 4.1.4 | Gradient Descent, Learning Rate and Optimization Functions | 45 |
| 4.2 | Convolutional Neural Networks..... | 47 |
| 4.3.1 | Convolution Operation and Feature Maps | 47 |
| 4.3.2 | Max Pooling, Stride and Padding | 48 |
| 4.3.3 | Weights Initialization..... | 50 |
| 4.3.4 | Normalization..... | 50 |
| 4.3.5 | Model Optimization and Regularization | 51 |
| 5 | Retinal Quality Assessment Experiments | 54 |
| 5.1 | Materials | 57 |
| 5.1.1 | Proprietary Dataset | 57 |
| 5.1.2 | Public Datasets | 58 |
| 5.2 | Methods | 60 |
| 5.2.1 | Data Preprocessing | 60 |
| 5.2.2 | Data Preparation | 64 |
| 5.2.3 | Convolutional Neural Networks | 67 |
| 6 | Results and Discussion | 76 |
| 6.1 | Classification Measures..... | 78 |
| 6.2 | Classification Results of Focus Assessment | 83 |
| 6.2.1 | Results of Model Net1 | 84 |
| 6.2.2 | Results of Model Net2 | 92 |
| 6.3 | Classification Results of Color Assessment | 100 |
| 6.3.1 | Results of Model Net1 | 101 |
| 6.3.2 | Results of Model Net2 | 107 |
| 6.4 | Classification Results of Illumination Assessment..... | 113 |
| 6.4.1 | Results of Model Net1 | 114 |
| 6.4.2 | Results of Model Net2 | 120 |

| | | |
|-----|---|-----|
| 6.5 | CNN Classification Performance for each Quality Parameter | 125 |
| 6.6 | Overall Quality Assessment | 126 |
| 7 | Conclusions..... | 131 |
| 7.1 | Conclusions..... | 133 |
| 7.2 | Future Work..... | 135 |
| | References..... | 137 |
| | Appendices | 146 |
| | A – Python Modules..... | 148 |
| | B – Distribution of Train, Validation and Test Sets..... | 152 |
| | C – CNN Structures | 157 |

LIST OF FIGURES

| | |
|--|----|
| Figure 2-1 Human eye anatomy (adapted from [4]) | 10 |
| Figure 2-2 The retina (adapted from [4]) | 10 |
| Figure 3-1 Example of retinal disc diameter (DD) and disc-macula distance (DM). Adapted from [16]. | 19 |
| Figure 3-2 Field definition comparison between 2 retinal images. | 19 |
| Figure 3-3 Examples of a good quality retinal image (upper left) and a bad retinal image (down left) and their corresponding edge distribution. The gray-scaled images were inverted for better visibility (adapted from [22]). | 23 |
| Figure 3-4 Scatter plot showing the separability of the three classes “Good image”, “Fair image” and “Bad image” (adapted from [22]). | 24 |
| Figure 3-5 Distribution of automatic quality evaluation grade values. AEQ = 4 (green) means highest quality and AEQ = 1 (red) means lowest quality (adapted from [8]). | 25 |
| Figure 3-6 A yellow grid is placed showing the seven regions where the features are calculated (adapted from [13]). | 26 |
| Figure 3-7 Digital retinal images with their FOV and noise mask (adapted from [12]). | 28 |
| Figure 3-8 Architecture of the shallow network (adapted from [24]). | 29 |
| Figure 3-9 Local and global saliency maps obtained through the methods [5-7] and the authors' method <i>Mahapatra</i> (adapted from [26]). | 31 |
| Figure 3-10 CNN architecture proposed in the present approach (adapted from [26]) | 32 |
| Figure 3-11 Learned filters form the last convolutional layer (adapted from [26]). | 32 |
| Figure 4-1 Learning curves with and without overfitting. Adapted from [32]. | 38 |
| Figure 4-2 Comparison between Machine Learning and Deep Learning. Adapted from [34]). | 39 |
| Figure 4-3 Progress chronology of Artificial Intelligence, Machine Learning and Deep learning concepts. Adapted from [35]). | 39 |
| Figure 4-4 Representation of a neuron in an artificial neural network Adapted from [33]. | 40 |
| Figure 4-5 Example of an Artificial neural network, with three layers. | 41 |
| Figure 4-6 Cross-entropy (log loss) when the true label is 1 and the predicted label is 1 too. | 42 |
| Figure 4-7 A perceptron representation with the weights, activation functions, the bias and the function σz | 43 |
| Figure 4-8 Softmax layer. Adapted from [38]. | 44 |
| Figure 4-9 Neural network activation functions. | 44 |

| | |
|--|----|
| Figure 4-10 Gradient Descent Algorithm 3D visualization. Adapted from [39]. | 45 |
| Figure 4-11 Loss curves of the different optimization functions. Adam is the optimization function with better learning, since the loss curve decreases continuously. Adapted from [40]. | 46 |
| Figure 4-12 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. | 48 |
| Figure 4-13 Convolution operation of input image I with kernel K , resulting $I * K$. | 48 |
| Figure 4-14 Max pooling operation. Adapted from [49]. | 49 |
| Figure 4-15 Padding operation in an input image. | 49 |
| Figure 4-16 Dropout regularization. Adapted from [53]. | 52 |
| Figure 5-1 Methodology pipeline applied to automated retinal image quality assessment using a Deep Learning approach. | 56 |
| Figure 5-2 Examples of images present in the APDP dataset. | 58 |
| Figure 5-3 Examples of retinal images from the IDRiD dataset (adapted from [55]). | 59 |
| Figure 5-4 Implemented preprocessing pipeline. | 60 |
| Figure 5-5 Result of the mask generation of the retinal images. | 61 |
| Figure 5-6 Result of the cropping process, (c) is obtained by adding a bounding box in the original image in (a), through an area obtained by the mask image in (b). | 62 |
| Figure 5-7 Result of the resizing process. The dimensions of the original image in (a) is 2560×1920 pixels; in the cropped image (b) 2306×1920 pixels and in (c) 512×512 pixels. | 63 |
| Figure 5-8 Diagram that explains the proportions of the train, validation and test sets split used in every CNN train and test. | 65 |
| Figure 5-9 Reading of a CSV file with Pandas DataFrame, in a Jupyter Notebook. | 66 |
| Figure 5-10 AlexNet architecture (adapted from [25]). | 67 |
| Figure 5-11 shallowNet architecture (adapted from [24]). | 67 |
| Figure 5-12 Net1 neural network. | 68 |
| Figure 5-13 Net2 neural network. | 68 |
| Figure 5-14 Weights and bias initializations. | 70 |
| Figure 5-15 Sequential model configuration of Net1. | 71 |
| Figure 5-16 Log file generated with the classification metrics. | 74 |
| Figure 6-1 ROC Curves. Curve A represents and $AUC=1$ and a perfect test; curve B are a good and a moderate diagnostic results, respectively; and D is the random classifier, with an $AUC=0.5$. Adapted from [76]. | 82 |

| | |
|---|-----|
| Figure 6-2 Accuracy and Loss learning curves of the trained network, with Batch normalization, for LR=0.0001 and the ADAM optimization function..... | 85 |
| Figure 6-3 Accuracy and Loss learning curves of the trained network, with Batch normalization and without the dropout layer, for LR=0.0001 and the ADAM optimization function. | 85 |
| Figure 6-4 ROC curve for the best classification model, with LR=0.001 and SGD optimization function. | 88 |
| Figure 6-5 Images that represent instances of wrong blurred predicted images. a), b), c), d) and e) are labeled as blurred but predicted as focused. However, it can be concluded that d) and e) were correctly classified by the network as focused and incorrectly manually classified by the expert. This was a success case in which the network learned the features that differentiate between focus and blur. .. | 88 |
| Figure 6-6 Images that represent instances of wrong focused predicted images. a), b), c) and d) are labeled as focused but predicted as blurred. It can be concluded that a) and d) are definitely focused, b) is a little blurred in the fovea area and c) is blurred and is a case of bad manual classification by the expert. Once again, the network learned the main differences between focused and blurred images. | 89 |
| Figure 6-7 Feature maps of each convolutional layer for Net1. | 90 |
| Figure 6-8 ROC curve for the best classification model, with LR=0.01 and SGD optimization function with $L2$. The ROC curve was generated with 3 points..... | 96 |
| Figure 6-9 Images belonging to the positive class (blurred images) that were classified as focused. In this case none of the images was correctly classified nor surpassed the human classification, since none of the images is focused. | 97 |
| Figure 6-10 Images belonging to the negative class (focused images) that were classified as blurred. Image a), b) and c) are focused while d) and e) were correctly classified by the network and incorrectly classified by the specialist. | 97 |
| Figure 6-11 Feature maps of each convolutional layer for Net2. | 99 |
| Figure 6-12 Learning curves for the model trained with Batch Normalization and without regularization $L2$ (left) and with regularization $L2$ (right). | 102 |
| Figure 6-13 Images that the network classified differently from the human. | 104 |
| Figure 6-14 Feature maps of each convolutional layer for Net1, of a bright test image. | 106 |
| Figure 6-15 Accuracy and Loss learning curves along the training, for the train and validation datasets, without regularization on the left and with regularization on the right. | 108 |
| Figure 6-16 Images with classification made by Net2 network, different from human classification. | 109 |
| Figure 6-17 Feature maps of each convolutional layer for Net2. | 110 |

| | |
|---|-----|
| Figure 6-18 Learning curves, of train and validation for the two trained models, only with Batch Normalization (left) and Batch Normalization+L2 regularization (right). | 115 |
| Figure 6-19 ROC curve for the best classification model, with LR=0.0001 and Adam optimization function. ROC curve was created with 3 points..... | 116 |
| Figure 6-20 Images belonging to the positive class (images classified by the human as uneven illumination) and that were classified as negative class (even) by the network. Images f) and g) were correctly classified by the network since their illumination is regular along all the retinal perimeter. These two cases exceeded the human classification, and the network generalized well for these images.. | 117 |
| Figure 6-21 Images belonging to the negative class (images classified by the human as even illumination) and that were classified by the network as positive class (uneven). The last image of the second line and all the images that belong to the last line of images are correctly classified as uneven by the network and incorrectly classified by the human classification..... | 117 |
| Figure 6-22 Feature maps of each convolutional layer for Net1, of an uneven test image. | 119 |
| Figure 6-23 ROC curve and AUC value. ROC generated with 3 points. | 121 |
| Figure 6-24 Images belonging to the negative class (images classified by the human as even illumination) and were classified by the network as positive class (uneven). Image d) contains artifacts and the network confused as a case of uneven illumination. Images g) and h) were correctly classified by the network as uneven and incorrectly classified by human classification..... | 121 |
| Figure 6-25 Images belonging to the positive class (images classified by the human as uneven illumination) and that were classified as negative class (even). Images f) to j) were correctly classified by the network since their illumination is regular along the perimeter of the retina. These two cases exceeded the human classification, and the network generalized well for these images..... | 122 |
| Figure 6-26 Feature maps of each convolutional layer for Net2. | 123 |
| Figure 6-27 Images that belong to the negative class (images classified by the human as “accept”) and that were classified by the network as negative class (“reject”). Image (d) and (e) were correctly classified by the network, since the image d) appears with blurred optic disc and vessels. Image e) is quite dark and contains a uneven illumination on most of the retinal border, so, is correctly classified by the network. | 127 |
| Figure 6-28 Images belonging to the positive class (images classified by the human as “reject”) and that were classified by the network as negative class (“accept”). The image c) was the only image correctly classified by the network in 4 images, since it contains good illumination, focus and a normal color, which allows to see all the details of the retina. | 127 |

| | |
|--|-----|
| Figure B-0-1 Distribution of focused and blurred retinal images in train dataset acquired from various sources..... | 152 |
| Figure B-0-2 Distribution of focused and blurred retinal images in validation dataset acquired from various sources..... | 152 |
| Figure B-0-3 Distribution of focused and blurred retinal images in test dataset acquired from various sources..... | 153 |
| Figure B-0-4 Distribution of train images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images)..... | 154 |
| Figure B-0-5 Distribution of validation images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images)..... | 154 |
| Figure B-0-6 Distribution of test images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images)..... | 154 |
| Figure B-0-7 Distribution of train images for each class – class 0 (even images), class 1 (uneven images). | 155 |
| Figure B-0-8 Distribution of validation images for each class – class 0 (even images) and class 1 (uneven images). | 155 |
| Figure B-0-9 Distribution of test images for each class – class 0 (even images) and class 1 (uneven images). | 155 |

LIST OF TABLES

| | |
|---|-----|
| Table 2-1 A summary of the principal retinal imaging modalities, including the method of image formation, typical resolution (in micrometers) and some of the advantages and limitations [6]. | 12 |
| Table 2-2 Findings assessed by graders in 3D OCT and Fundus Images. | 14 |
| Table 3-1 Classification accuracy results on the test set for five classification models (adapted from [24]). | 30 |
| Table 3-2 Classification results for different methods compared to CNN (adapted from [26]). | 33 |
| Table 5-1 Specifications of the experiments computer. | 57 |
| Table 5-2 Summary of all the parameter values of CCN topology. | 70 |
| Table 6-1. Confusion matrix for binary classification. | 79 |
| Table 6-2. Confusion matrix for multi-class classification. | 79 |
| Table 6-3 Fixed parameters used in each model Net1 and Net2. | 83 |
| Table 6-4 Varied parameters used in each model Net1 and Net2. | 83 |
| Table 6-5 Classification performance values for each model trained with each LR and optimization function, without Batch Normalization. | 84 |
| Table 6-6 Training performed for the LR = 0.0001 and the ADAM optimization function, where A represents the training without BN, B represents training with BN and C represents the training with BN and without dropout. | 86 |
| Table 6-7. Final frame with the all the best classification metric values for each LR and optimizer studied. With a star are marked the models that performed better without Batch Normalization. | 86 |
| Table 6-8 Results obtained from the best classifier trained without Batch Normalization (A) and with Batch Normalization (B). | 87 |
| Table 6-9. Confusion matrix for the best focused images classifier. | 87 |
| Table 6-10 Comparison of results before and after recalculation of all classification metrics. | 89 |
| Table 6-11 Learning curves of models trained with Batch Normalization and L2 regularizer, for two different LR and SGD optimization functions (trained with and without momentum). | 93 |
| Table 6-12 Results obtained without the use of regularization $L2$. | 95 |
| Table 6-13 Results obtained with the use of regularization $L2$. | 95 |
| Table 6-14 Confusion matrix for the best focused images classifier, with LR=0.01, SGD optimization function and with $L2$. | 96 |
| Table 6-15 Comparison of results before and after recalculation of classification metrics. | 98 |
| Table 6-16 Fixed parameters used in each model Net1 and Net2. | 100 |

| | |
|--|-----|
| Table 6-17 Varied parameters used in each model Net1 and Net2. | 101 |
| Table 6-18 Results obtained from the classification of color images of Net1 network..... | 101 |
| Table 6-19 Results of the classifier performance for LR = 0.0001 and SGD optimization function with momentum, using Batch Normalization, different batch sizes and the use of $L2$ regularization. | 102 |
| Table 6-20 Confusion matrix for the best result obtained from Net1 network, with Batch Normalization and $L2$ regularization. | 103 |
| Table 6-21 Comparison of results before and after recalculation of classification metrics. | 105 |
| Table 6-22 Results of the color image classification of Net2. | 107 |
| Table 6-23 Results of images classification, with the $L2$ regularization implementation, in Net2..... | 107 |
| Table 6-24 Confusion matrix with the distribution of the classification given by Net2 network. | 108 |
| Table 6-25 Comparison of results before and after recalculation of classification metrics. | 111 |
| Table 6-26 Fixed parameters used in each model Net1 and Net2..... | 113 |
| Table 6-27 Varied parameters used in each model Net1 and Net2. | 113 |
| Table 6-28 Results obtained from the illumination classification of Net1. | 114 |
| Table 6-29 Comparison of the results obtained by the trained model with and without regularization. | 116 |
| Table 6-30 Confusion matrix with the distribution of the classification given by Net1 network. | 116 |
| Table 6-31 Comparison of results before and after recalculation of classification metrics. | 118 |
| Table 6-32 Results obtained from the illumination classification of Net2 network. | 120 |
| Table 6-33 Confusion matrix with the distribution of the classification given by Net2 network, for LR=0.0001 and Adam..... | 121 |
| Table 6-34 Comparison of results before and after recalculation of classification metrics. | 122 |
| Table 6-35 Results of the focus classification of Net1 and Net2 networks. | 125 |
| Table 6-36 Results of the color classification of Net1 and Net2 networks. | 125 |
| Table 6-37 Results of the illumination classification of Net1 and Net2 networks..... | 125 |
| Table 6-38 Fixed parameters used in model Net1. | 126 |
| Table 6-39 Results obtained with the parameters of best focus assessment model. | 126 |
| Table 6-40 Confusion matrix of Net1. | 127 |
| Table 6-41 Results obtained for each case. Case A corresponds to the results without the labels correction and case B to the labels correction. | 128 |
| Table 6-42 Accuracy, Sensitivity and Specificity values for different state-of-the-art approaches and the proposed method. With dashed lines, are the values unknown or not mentioned in the documents. | 128 |

LIST OF ABBREVIATIONS

A

Adam Adaptive Moment Estimation

AI Artificial Intelligence

ANN Artificial Neural Network

AUC Area under the curve

B

B Batch Normalization

C

CNN Convolution Neural Networks

CM Confusion Matrix

CSV Comma-Separated Values

CUDA Compute Unified Device Architecture

D

DL Deep Learning

DME Diabetic Macular Edema

DR Diabetic Retinopathy

F

FCL Fully Connected Layer

FN False Negative

FP False Positive

FPR False Positive Rate

G

GPU Graphics Processing Unit

I

IQA Image Quality Assessment

J

JPEG Joint Photographic Experts Group

L

LR Learning Rate

M

ML Machine Learning

MLP Multi-Layer Perceptron

N

NN Neural Networks

O

OS Operating System

P

P Precision

R

ReLU Rectified Linear Unit

RMSProp Root Mean Square Propagation

ROC Receiver Operating Characteristic

S

SGD Stochastic Gradient Descent

T

TN True Negative

TP True Positive

TPR True Positive Rate

V

VM Virtual Machine

GLOSSARY

| | |
|------------------------------------|---|
| Activation Function | Used in Neural Networks, this is the function in the neuron, that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically non-linear) to the next layer. |
| Artificial Intelligence (AI) | Computer programs designed that allows to make decisions and perform tasks that simulate human intelligence, human brain and behavior. |
| Artificial Neural Networks (ANN) | A model that, developed by the human brain inspiration, is composed of input layers, hidden layers and output layers of simple connected units or neurons followed by activation functions. |
| Batch | In Neural Networks, it is a set of examples or samples used in one iteration/epoch. |
| Class | In ML, a class refers to the output category of the data. A label in a dataset points to one of the classes. |
| Convolutional Neural Network (CNN) | A CNN is a deep neural network that is currently the state-of-the-art in image processing, in the recent years. Its major advantage is the little pre-processing steps required when compared to other image classification algorithms. |
| Deep Learning (DL) | A subset of AI and Machine learning which allows multi-processing layers and computational models to learn representations of data with multiple levels of abstraction. |
| Epoch | A full training iteration over the entire data set such that each example has been seen once. |
| Overfitting | Phenomenon that occurs when training the model, training data is memorized by the model and, then can't generalize in new unseen data |
| Test set | The subset of the dataset that is used to test the model after the model has gone through initial vetting by the validation set. |
| Train set | The subset of the dataset used to train a model. |
| Validation set | A subset of the dataset — disjunct from the training set—that is used to adjust hyperparameters. |
| Dropout | A form of regularization in training neural networks. This regularization removes random selection neurons, by a given fixed parameter p . The more units dropped out, the stronger the regularization. |

1 INTRODUCTION

1.1 MOTIVATION

The diagnosis of ocular diseases, such as Diabetic Retinopathy (DR) and Diabetic Macular Edema (DME) heavily relies on imaging techniques. One of the most used eye imaging modalities is fundus eye imaging because it is non-invasive and has low operating costs.

At the time of image acquisition for eye diseases diagnosis, several problems may arise, such as the acquisition of low-quality images, where the lesions may become nebulous, leading to classification errors.

To better understand the underlying causes and progression of DR and DME, the research community needs to analyze in detail, large amounts of retinal images over a long period of time, and a study based only on human classifiers is a time-consuming and error-prone task, since the experience, visual perception, types of cameras and judgment of photographers can vary [1],[2].

Another disadvantage of operator-dependent classifications are the increased rate of poor image acquisitions and repeated ophthalmology examinations, which has associated financial and time costs.

To overcome these limitations, it is proposed the study of an automatic image quality assessment classifier, which aims to classify with low computational resources and in short time. This classifier would receive as input a varied set of images, returning as outputs, the images that are of good quality and those of poor quality, depending on the focus, the color and the illumination of these.

It is intended that the developed classifiers would be able to learn the necessary features to generalize well in new unseen images and classify with a low number of false negatives and false positives detections.

1.2 OBJECTIVES AND RESEARCH QUESTIONS

The present work aims to design and develop reliable and fast algorithms that can evaluate the quality of the images, according to their focus, illumination and color parameters present in them. It is also intended that the algorithm can make an overall assessment of each image and that it outputs whether it accepts or rejects the image or set of images.

Throughout this study, it is intended to find the answer to the following research questions:

- What are the characteristics that contribute to the evaluation of image quality?

- How well can the model handle the use of different values and types of manually selected hyperparameters based on theoretical and empirical knowledge?
- Are the models learning with different parameters and methods applied? Which techniques increase the accuracy and performance of a model?
- Are the implemented models, computational and time expensive? Which model performs best?
- Which characteristics highlights the present work from the previously works made? Which are the limitations of the present work?
- Can these models be compared with the state-of-the-art techniques?

1.3 RESEARCH METHODOLOGY

In the present work, the initial phase contemplates the choice of the theme, the motivation, the definition of objectives and the creation of a plan of activities for the development of the study, in order to respond to the previous objectives.

The next phase consisted on researching relevant sources of information such as articles and books, where these concepts and the use of these works were constantly renewed as new ideas or information emerged.

To find the necessary knowledge to carry out the present work, the research was based on the use of keywords in the most prestigious editors and digital libraries of medical and scientific literature, such as: Elsevier, Springer, IEEE Xplore, PubMed, ScienceDirect, Google Scholar, ResearchGate, ACM Digital Library, SPIE Digital Library among other sources.

Some of the keywords presented were used in the research:

- Retinal Quality Assessment;
- Retina;
- Fundus Photography;
- Diabetic Retinopathy;
- Diabetic Macular Edema;

- Deep Learning;
- Convolutional Neural Networks;

The articles that did not show results with these or other similar, work-related keywords were excluded.

In the final part of this work, it was proceeded the development and analysis of functional systems able to reach the proposed objectives.

The applied methodology comprises the following stages of development:

- Definition of the problem, its motivation, and its characteristics;
- Elaboration of the state of the art and objectives of the work;
- Development of algorithms that allow achieving the proposed objectives, as well as improvements and corrections of these based on the obtained results;
- Presentation of conclusions, a discussion of the results and proposal of future work.

1.4 DISSERTATION STRUCTURE

This work is structured in 7 chapters including this as a first chapter which frames the reader with the context, motivations and objectives of the study. Next, the second chapter is presented with the medical background, with all the important concepts associated to the diseases and anatomy present in the eye. The third chapter includes the information about each of the retinal quality parameters and the state-of-the-art of the past 20 years in retinal image quality assessment methods. In chapter 4, Machine Learning, Deep Learning and image processing concepts are presented, to better understand the pipeline developed and techniques used in the chapter 5. This chapter also presents which types of images and datasets used in the classification CNN algorithms. In chapter 6 the results and discussion of the networks developed are presented and in chapter 7, the main conclusions and the future work.

2 MEDICAL BACKGROUND

2.1 EYE ANATOMY

The eye is a complex optical structure that is able to reflect and focus light that stimulates neural responses. The eye is essentially made up from a number of optical components, neural components, and supportive layers, as can be shown in Figure 2-1. At the front of the eye, a thin and transparent membrane known as the cornea, covers the anterior surface of the eye. This membrane has a dual purpose eye protection and refracting the light that enters in the eye. A portion of the light passing through the cornea passes through the pupil, a small opening in front of the lens.

The choroid contains a network of blood vessels that serve as the major source of nutrition to the eye and is a membrane that lies directly below the sclera. The choroid coat is heavily pigmented, helping to reduce the amount of extraneous light entering the eye and the backscatter within the optic globe.

The lens is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It contains 60 to 70% water, about 6% fat and more protein than any other tissue in the eye.

The innermost membrane of the eye is the retina, shown in Figure 2-2, which lines the inside of the wall's entire posterior portion. When the eye is properly focused, light from an object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina. There are two classes of receptors: cones and rods. The cones in each eye number between 6 and 7 million. They are located primarily in the central portion of the retina, called of the fovea, and are highly sensitive to color. Humans can resolve fine details with these cones largely because each one is connected to its own nerve end [3].

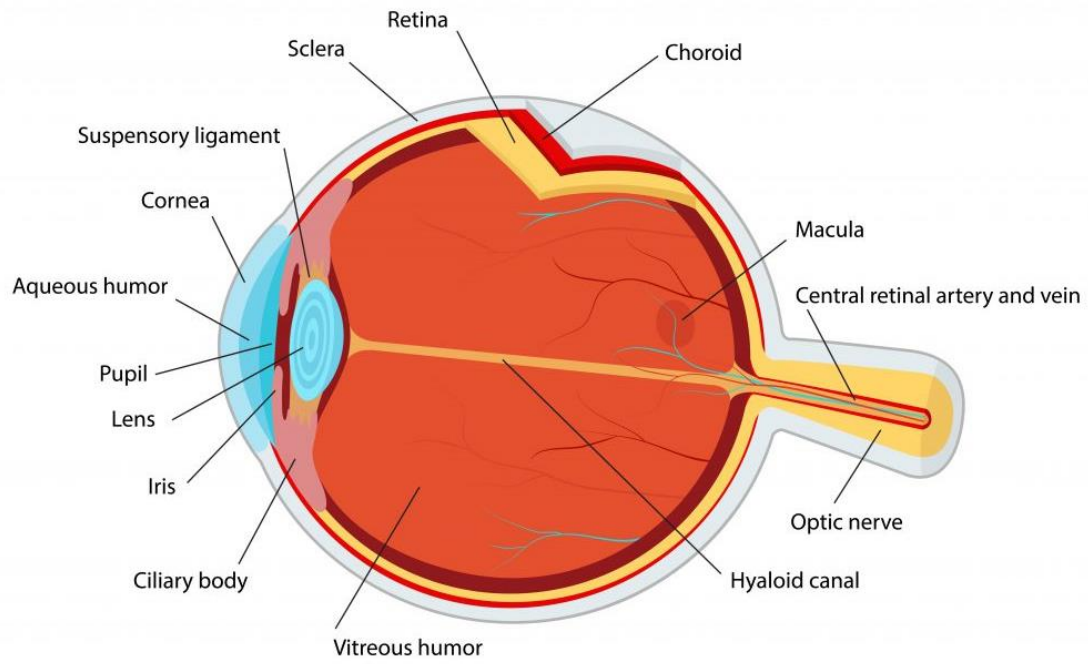


Figure 2-1 Human eye anatomy (adapted from [4])

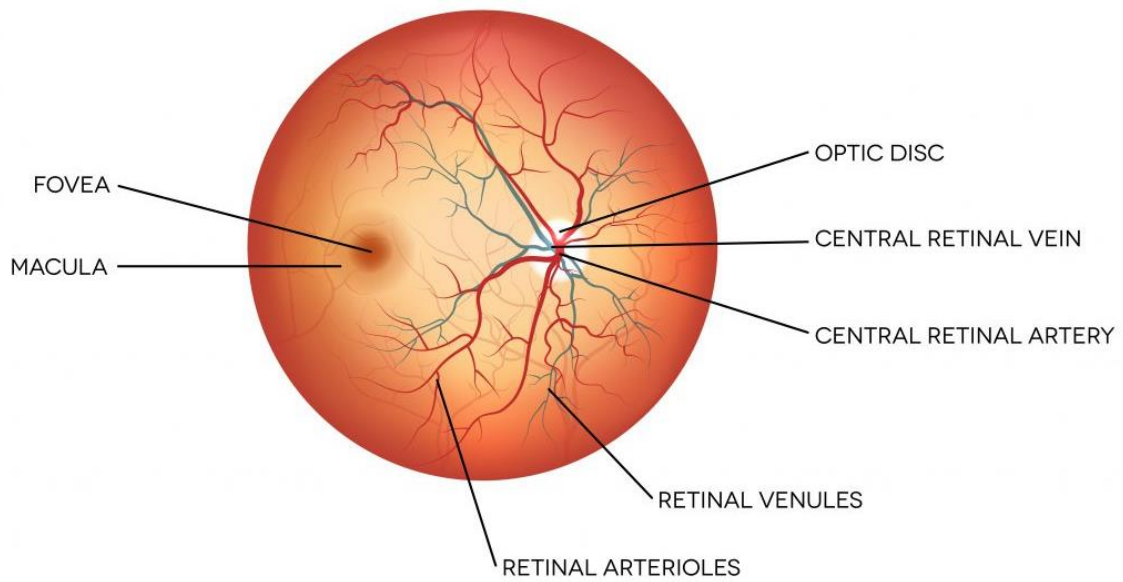


Figure 2-2 The retina (adapted from [4]).

2.2 RETINAL IMAGING MODALITIES

Retinal Imaging has developed rapidly during the last 160 years and is now a mainstay of the clinical care and management of patients with retinal as well as systemic diseases [5].

In the ophthalmology clinic, retinal imaging devices are primarily used in the diagnosis of retinal disease as well as a serial monitoring in retinal conditions such as age-related macular degeneration to monitor response to treatment. However, the detail with which the eye can be visualized non-invasively opens up investigative possibilities for a variety of long-term conditions.

Fundus imaging is defined by [5] as the process which results in a 2D image, where the image intensities represent the amount of a reflected quantity of light [5].

Fundus imaging generates a two-dimensional (2D) image of the interior three-dimensional (3D) surface of the eye and is performed with a system that consists of a specialized low-power microscope and an attached camera. The patient sits with his/her chin in a rest and forehead placed against a bar, while the operator focuses and aligns the camera before pressing the shutter release to fire a flash and create the image. This image is an upright, magnified picture of the fundus with typical angles of view of 30, 45 or 60 and with a magnification of x2.5, depending on the system optics. Images of higher quality can often be achieved by dilating the pupils beforehand with mydriatic eye drops to enlarge the FOV of the fundus and improve image quality. Current digital image resolutions are around 3000 x 3000 pixels [6].

Fundus photography is widely used for population-based, large-scale detection of some retinal diseases like diabetic retinopathy, glaucoma, and age-related macular degeneration (AMD). Optical coherence tomography and fluorescein angiography are widely used in the management and diagnosis of patients with DR, AMD, and inflammatory retinal diseases [5].

In Table 2-1 is presented a summary of the principal retinal imaging modalities and their most differences.

Table 2-1 A summary of the principal retinal imaging modalities, including the method of image formation, typical resolution (in micrometers) and some of the advantages and limitations [6].

| Modality | Image formation | Resolution (μm) | Advantages | Limitations |
|-------------------------------|--|------------------------------|--|--------------------------------------|
| Fundus camera | Colour photograph of retinal surface | 7–20 | Blood vessels—marker of microvascular health | Dilation of pupils often needed |
| | | | Lesions, exudates, haemorrhages—common signs of diabetic retinopathy | |
| Scanning laser ophthalmoscope | Focused laser beam scans retinal surface | 10–15 | Larger field of view in one image than fundus camera | Typically more expensive and complex |
| | | | Examine fundus features in the peripheral retina | |
| Angiography | Retina illuminated at wavelength to excite fluorescent dye | 7–20 | Improved contrast and detail of vasculature | Invasive—intravenous injection |
| | | | Flow velocity, leakages, blockages—markers of microcirculatory health | |
| Optical coherence tomography | Near-infrared light penetrates retina; interferometry resolves tissue layers | 4 | Cross-sectional view of internal retinal structures | Susceptible to media opacities |
| | | | Changes in the retinal nerve fibre layer—possible markers of neurodegeneration | |

2.1.1 DIGITAL FUNDUS PHOTOGRAPHY

Automated diagnosis of retinal fundal images using digital imaging analysis offers huge potential benefits. In a research context, offers the potential to examine a large number of images with cost and time savings and offer more objective measurements the current observer-driven techniques [7].

The use of digital imaging systems has reduced the technical failure rate and the electronic image has been facilitating easy storage and cataloging. These digital systems for retinal photography acquisition have been shown to achieve sensitivities and specificities of approximately 90% in detecting referable DR. Comparisons between film and digital fundus photographs found agreement to be almost perfect for DR severity level and moderate to substantial for DME [8].

2.1.2 FLUORESCEIN ANGIOGRAPHY

Fluorescein angiography has been utilized to provide measurements of overall fluorescein intensity variation over the fluorescein transit, or to examine vascular structures in the eye such as the choroid, iris and retina. A fluorescein dye is injected and remains within normal blood vessels, thus leakage into surrounding tissue indicates vascular pathology. Fluorescein angiography can offer a

potential diagnostic index of retinopathy severity and it is most commonly used to investigate retinal diseases such as diabetic retinopathy.

Fluorescein is a vital dye; it is composed of crystalline hydrocarbon dye and has the property of absorbing light in the blue wavelength and emitting it in the green wavelength [7].

This techniques also have drawbacks associated: it is an prohibit intervention in large-scale screening analysis and can contribute to nausea and vasovagal attack. The mortality associated with this technique is 1 in 200 000 patients. Its use should be limited to diagnosis in patients whose management may be altered by the results or subjects included in an ethically approved research study [9].

2.1.3 OPTICAL COHERENCE TOMOGRAPHY

OCT is also widely used in preparation and follow-up in vitreoretinal surgery [5].

Optical coherence tomography (OCT) is a non-invasive, noncontact transpupillary imaging modality that has offered revolutionized ophthalmic clinical practices. This technique consists in using light with low coherence interferometry. OCT produces cross-sectional images of the macula allowing objective evaluation of macular thickness and evaluation of vitreomacular interface. However, it gives a poor correlation between macular thickness and visual acuity. The main disadvantage of OCT is the cost of the required equipment, which limits its availability [10].

This technique has the following procedures: an optical beam is directed at the target tissue and interferometry resolves the back-scattered light signals. The scanning beam is split with a beam splitter sending some to the target tissue (target arm) and the remaining portion to a reference mirror (reference arm). Both beams are reflected back to the beam splitter from the target and the reference mirror, respectively, and then directed together to a detector. When the distance to the reference mirror in the reference arm is equal to the distance to the reflecting target within the tissue, interference occurs, inferring to the depth of the reflecting structure in the target [8].

In Table 2.2, its presented the main differences found in the OCT and Fundus Imaging modalities.

Table 2-2 Findings assessed by graders in 3D OCT and Fundus Images.

| | 3D-OCT Findings | Fundus Image Findings |
|---------------------------|--|---|
| Epiretinal | ILM Irregularity | EMR Macular Hole |
| Retinal/Subretinal | Increased Retinal Thickness Decreased Retinal Thickness Hyperreflective Features Hyporefective Features | Microaneurysms Cotton Wool Spots Exudate Hemorrhage Pigmentary Change |
| RPE/Choroidal | RPE Irregularity | Drusen RPE Atrophy |

3 RETINAL IMAGE QUALITY ASSESSMENT

Retinal images can be acquired through Fluorescein Angiography, Optical Coherence Tomography, and Digital Fundus Photography as discussed in Chapter 2. Before proceeding to the diagnosis, the image quality is assessed by the physician since it can impact a correct image analysis. Several parameters affect retinal image quality and its classification is usually manual. This rating is a slow, highly subjective and error-prone process even if rated by experienced professionals. In order to make the whole process more efficient, it is necessary to rely on an automated analysis and giving results in a short time can reduce the time of evaluation, the workload of the specialists, as well as reduce the discomfort of rescheduling further examination of patients [7]. In this chapter the image quality parameters regarding retinal fundus images will be described. In addition, an analysis of the state of the art methods that address this issue will be presented as well.

3.1 RETINAL IMAGE QUALITY PARAMETERS

Image sharpness is mainly influenced by camera focus at the moment of exposure, but also by any factor that has a blurring effect on the generated image (e.g. eye movement or cataract). Illumination reflects how well illuminated the retina is at exposure (that it is neither under or overexposed). Illumination is influenced by the flash settings of the camera, the pupil size and the pigmentation of the fundus [8].

Reliable factors for image quality assessment (IQA) identified by the Atherosclerotic Risk in Communities (ARIC) [9] study are grouped into two major categories: generic image quality parameters (e.g. contrast, clarity) and structural quality parameters (such as visibility of the optic disc and macula) [10].

Image quality is a difficult and subjective task in any field. Insufficient quality in medical images can affect the clinicians' capacity to perform a correct diagnosis. In general, depending on the use of the images the interpretation of quality can vary [11].

Subjective quality can also be measured by psychophysical tests or questionnaires with numerical ratings, but this is not the ideal type of evaluation when the immediate assessment is desired.

Quality of fundus images is usually verified by the photographer in the acquisition moment, and these images should be retaken if the image quality can impair an adequate assessment of key features in the retina. To capture high-quality fundus image, a proper camera-to-eye distance should be maintained to avoid haziness and artifacts, and also flash and gain should be adjusted to avoid severe over and underexposures.

Fundus Images are characterized by a dark background surrounding the FOV and cropping out the retinal area (do preprocessing in the images), diminishes the number of operations because it excludes the pixels outside the FOV [12].

Like *Davis et al.* [13], said in their study, several published clinical reports indicate that from 10% to 15% of fundus images are rejected from studies due to image quality.

Obtaining the highest possible image quality is very important when photographing a patient's retina in a clinic or collecting images of a subject for a study, because in longitudinal studies where every examination and image is critically important, losing an image for a given period may result the reduction in statistical power of results and, at worse, the loss of that individual diagnosis from the study. When the image is unacceptable, the best procedure to make, is retake the image(s), in order to reduce the inconvenience to a patient who will have to return to the clinic for re-imaging [13].

3.1.1 FIELD DEFINITION

The field definition parameter is evaluated by the reader according to the correct positioning and presence of the optical disc and the macula.

According to the Health Technology Board for Scotland [14] and *Fleming et al.* [15], fundus photographs should be centered on the macula and contain the optic disc.

The field definition is classified as:

- **Excellent:** The entire macula and optic disc are visible. The macula is centered horizontally and vertically in the image;
- **Good:** The entire macula and optic disc are visible. The macula is complete but is not centered horizontally and vertically in the image, with both main temporal arcades completely visible;
- **Inadequate:** A small artifact is present or at least the macula, optic disc, superior temporal arcade, or inferior temporal arcade are incomplete.

For the classification of diabetic retinopathy and diabetic macular edema, it is necessary that the fovea is at least 2 disc diameter (DD), from the edge of the image. The optic disc and temporal arcades must be complete and visible in the image because they are considered a guide to ensure that the photograph has been correctly aligned. The visual definition of DD is shown in Figure 3-1. Images in Figure 3-2 are a) with excellent field of view and b) inadequate field of view.

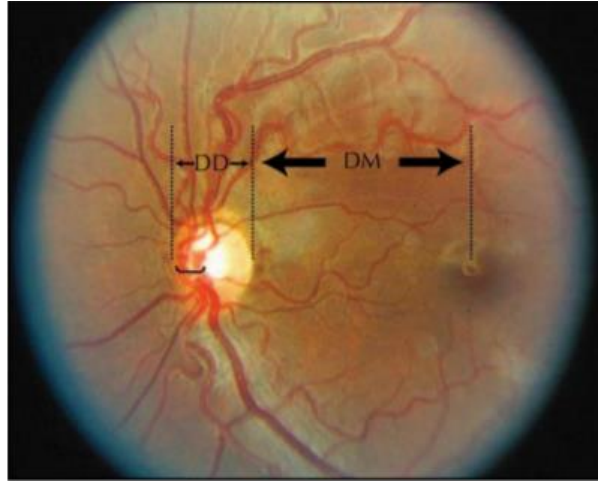


Figure 3-1 Example of retinal disc diameter (DD) and disc-macula distance (DM). Adapted from [16].

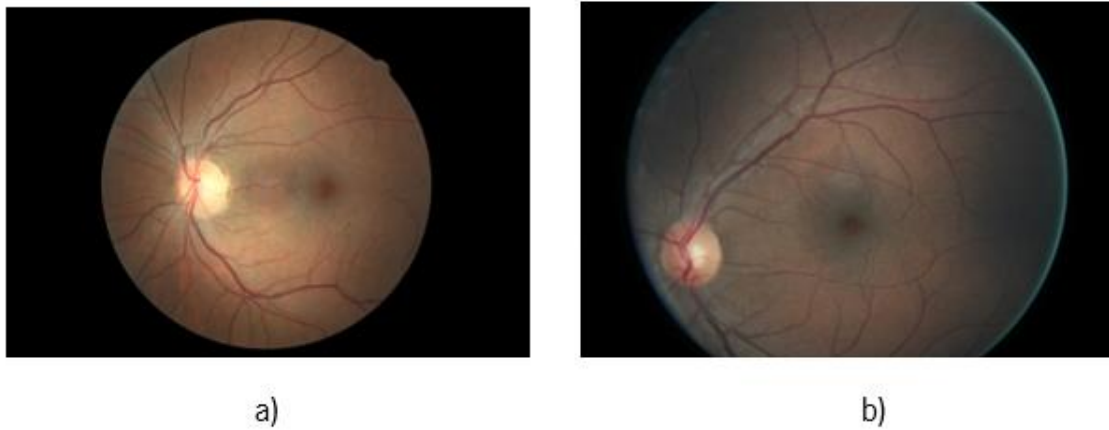


Figure 3-2 Field definition comparison between 2 retinal images.

3.1.2 CLARITY AND FOCUS

An image is classified as defocused or with poor clarity when the expert concludes, in the time of image acquisition, that an automatic system will be unable to accurately detect the main retinal features such as blood vessels or the presence of diabetic lesions [12], since when the blur is significant, it may misclassify the disease and classify it as a normal patient. This blur may be due to the presence of cataracts, the appearance of diabetic macular edema, poor camera focus, optics, or saccadic eye movement during acquisition [15].

The focus is classified as:

- **Excellent:** Small vessels are clearly sharp and visible within 1 DD around the macula, and the nerve fiber layer is visible;

- **Good:** Small vessels are clearly visible but not sharp within 1 DD around the macula or the nerve fiber layer is not visible;
- **Fair:** Small vessels are not clearly visible within 1 DD around the macula but are of sufficient clarity to identify third-generation branches within 1 DD around the macula.
- **Inadequate:** Third-generation branches within 1 DD around the macula cannot be identified.

Therefore, an image with adequate clarity is defined as one that shows sufficient details for the automated classification of the retinal quality assessment. The visibility of macular vessels have been used as an indicator of image clarity, since these vessels are known to be narrow and to become less visible with any image degradation [15].

3.1.3 VISIBILITY OF THE MACULA

To evaluate this parameter, it's analyzed if any portion of the macula is obscured by a dark shadow resulting from poor pupillary dilation or omitted from the image due to poor field definition. If the macula is visible, the image is assessed as pared, but if the macula is obscured, missing or has some kind of artifact present in it, is classified as impaired [9].

3.1.4 VISIBILITY OF THE OPTIC DISC

To evaluate the visibility of the optic disc, it's analyzed if any portion of the optic disc is obscured by an artifact, such as a dark shadow or a blink or even omitted from the image due to poor field definition. If the optic disc is visible, it is assessed as pared, but if the optic disc is obscured, missing or has some kind of artifact present in it, is classified as impaired [9].

3.1.5 ARTIFACTS

Retinal fundus cameras, just like any imaging device, suffer from particles or blemishes like dust particles attached to the sensor and lens. These particles may difficult its diagnostic purpose and reduce the image acuity and clarity. For example, these artifacts can be mistaken as small lesions, such as microaneurysms for the diagnosis of DR [17].

According to [9] there are some commonly artifacts present in the retinal images such as:

- **Dust and Dirt:** the presence of these sort of artifacts appear as white dots or spots that may have a varied size and are in the same location of the image, no matter which field of the retina is imaged, causing dirty lenses on the camera;
- **Lashes/Blink:** Lashes are characterized as a partial blink. Often appear on the bottom of the image as either light or dark linear “shadows”. These “shadows” can easily obscure the lower half of the image and, occasionally they appear in the upper half of the image as a bright reflectance, but doesn’t affect the ability to grade DR and DME lesions;
- **Arcs:** A patient with an incorrect distance from the camera or a small pupil may cause the appearance of arcs in the image, with yellowish, orange or bluish colors, in the size of a small slice or arc that obscures more than half the of the retina field. These artifacts may occur anywhere in the field, but they occur more frequently along the nasal or temporal margins;
- **Haze:** Haze is a spectral reflectance located centrally and can be seen along the periphery of the image. This artifact is the result of an excessive camera to the patient distance [2]; Two types of haze can be considered: an overall and edge haze. The overall haze reduces the clarity and generally produces a yellowish or greenish color over the retinal image, and a edge haze is generally white and most opaque at the periphery and diffusing towards the center of the image.

3.2 RETINAL IMAGE QUALITY ASSESSMENT STATE-OF-THE-ART

Several approaches and algorithms have been developed to automatically determine the quality of the retina, relying on generic image quality parameters, structural parameters, both generic and structural image quality parameters (hybrid parameters) and finally based in Deep Learning approaches. The structural and hybrid parameters-based approaches are out of the present study scope.

Based on generic IQA parameter methods are the following parameters: sharpness, illumination, contrast, focus, texture and color and are methods that use simple image measurements [18].

The methods based on structural IQA parameters require the identification and segmentation of anatomical structures of the retina such as the macula, optic disc, detection of small vessels around the fovea and retinal vessels. These approaches tend to lose robustness when the images

used for retinal structures segmentation are of poor quality [13], usually more complex and time-consuming than generic parameter approaches [19].

The combination of generic and structural IQA parameters is based on retinal vessel segmentation to obtain values of vessel density, based on histograms that extract features such as contrast, brightness and texture of retinal images [20].

The last approaches discussed are Deep Learning IQA approaches.

3.2.1 GENERIC IQA APPROACHES

Several methods have been developed to assess the retinal image quality with generic features.

Lee et al. [21], in 1999, were the first authors to develop an automated retinal image quality assessment method based on global image intensity histogram. They define a template intensity histogram whose parameters were derived from the analysis of 20 images with very good quality from a total of 360.

The parameters studied for each image were brightness, contrast and signal-to-noise ratio (SNR). Each histogram of the target image is compared to the template intensity histogram. To obtain the quality index Q (which determines the image quality), the target images were normalized, and subsequently, convolved with the template intensity histogram. The index Q took values between 0 and 1, being 0 the minimum value and 1 the maximum value taken; when Q was very close to 0, it meant that the image had low or very poor quality.

In 2001, *Lalonde et al.*, [22] proposed a region-based approach on the histogram/distribution of the edge magnitudes in the image and the local distribution of the pixel intensity, as opposed to the global histogram of *Lee et al.*

Lalonde et al. used a total of 40 retinal images, to study the following two criteria:

- d_{edge} – a measure of match between the edge magnitude distribution of the image, giving a global histogram.
- $d_{intensity}$ – a measure of match between intensity distributions of some regions of the image. This is an illumination measure, where a good retinal image should not have too many dark or white pixels.

They have noticed that the distribution of the edge magnitudes d_{edge} , in a good image, has a shape that is similar to Rayleigh distribution but with a gentler drop as the intensity increases. Images with bad quality show a Rayleigh distribution with an abrupt drop, as shown in Figure 3-3. The measure d_{edge} is also a focus measure because they experimented defocusing an image using a Gaussian or a mean filter and the more d_{edge} increases, more visible gets the blurring effect.

These two criteria were assessed to decide whether the images studied were of good, fair or bad quality, so they build edge and intensity models. Figure 3-4 shows points that are the outcome for assessing the quality of all the images and the symbols (+, × or o) were the specialist assessment. It is important to note that very few images were used in the study, and for better evaluation, more images should have been used and ideally a comparison with many human observers.

These authors also showed that the contribution developed by *Lee et al.*, between image quality and histogram similarity is not that strong. They noticed that some poor quality images had a histogram that resembled the template intensity histogram and also found good quality images with different histograms.

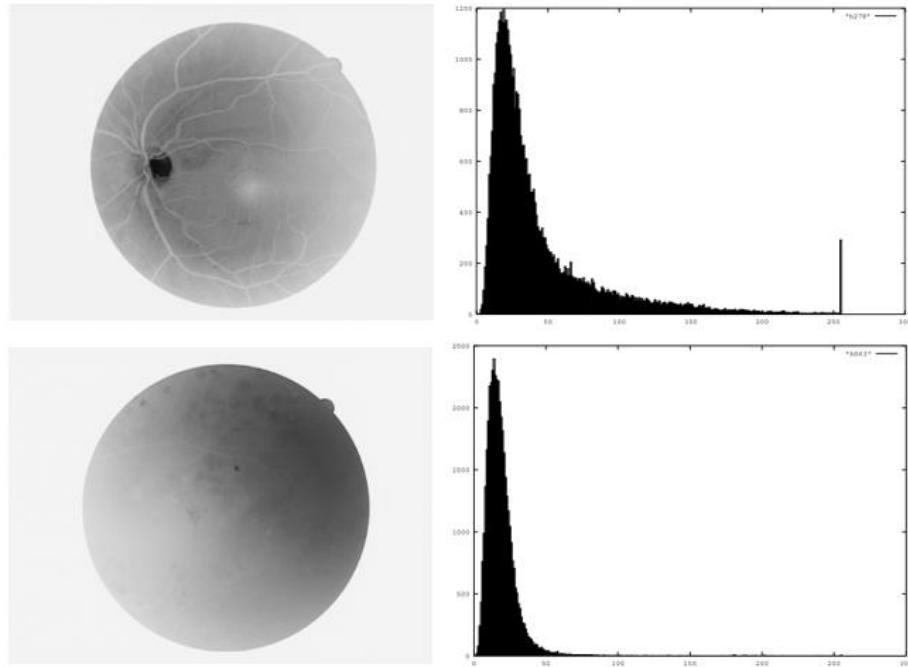


Figure 3-3 Examples of a good quality retinal image (upper left) and a bad retinal image (down left) and their corresponding edge distribution. The gray-scaled images were inverted for better visibility (adapted from [22]).

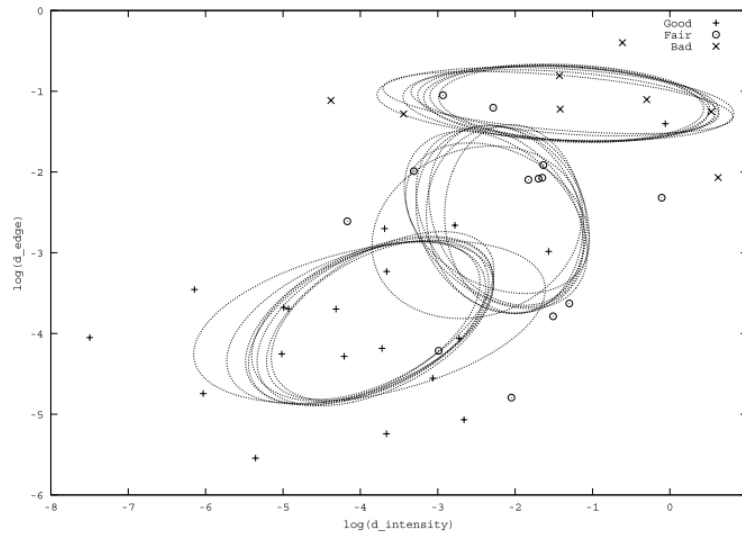


Figure 3-4 Scatter plot showing the separability of the three classes “Good image”, “Fair image” and “Bad image” (adapted from [22]).

The major inconvenience in these two histogram-based methods, developed by *Lee et al.* and *Lalonde et al.* were the small set of very good quality images used to construct the template histogram model and its limited type of analysis, since it doesn't consider the natural variance found in retinal images.

Bartling et al. [8], in 2009, developed a method based on sharpness and illumination features. A total of 1000 fundus images, randomly selected, were assigned to four quality classes (not accepted, accepted, good and very good). The images studied were divided into squares (64x64 pixels) and each square was analyzed separately.

The amount of structural content within each square was determined and only the squares containing sufficient structural content were used for the measurement of sharpness. The structural content was evaluated by convolving the sub-image using a Laplacian Operator, calculating the standard deviation of the pixel values and then analyzing the high-frequency magnitudes using the two-dimensional discrete Wavelet Transform, where higher frequency meant more sharpness.

As in sharpness, the measurements of the illumination parameter were performed after the image normalization and image square divisions. Measurements were first performed on the individual squares and then combined into a single value for the entire image. This single value was obtained by the fraction of all squares in an image classified as acceptable. The final illumination value for the entire image would be between 0 and 1.

With its method, *Bartling et al.* reached a concordance between the computed and human quality score, following a kappa approach, obtaining a median kappa value of 0.64 in a range of [0.52,0.68]. However, an unweighted kappa value approach (no image other than one category) was also used to analyze a concordance among all observers, obtaining a median kappa value of 0.55. Therefore, although the results were not excellent, the work developed by the authors shows that an automatic retinal image quality test is executable and more objective than human classifications.

Figure 3-5 shows the quality grade distribution values in 1000 fundus test images. These values are the product between the two properties (score = sharpness value x illumination value) and the lines indicate the borders between the quality groups, were score: > 10 (very good), > 6 and ≤ 10 (good); > 2 and ≤ 6 (acceptable) and ≤ 2 (not acceptable).

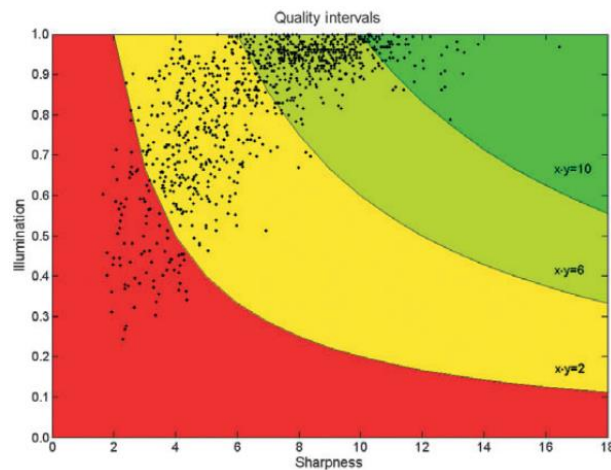


Figure 3-5 Distribution of automatic quality evaluation grade values. AEQ = 4 (green) means highest quality and AEQ = 1 (red) means lowest quality (adapted from [8]).

Also, in 2009, *Davis et al.* [13], developed a quality assessment method based on contrast and luminance features, using simple measures and keeping calculation time as low as possible.

In this study, the authors, calculated a total of 17 features, for each RGB color channel (R – Red channel, G – Green channel, B – Blue channel) and CIELab¹ in all the images studied.

The features were produced for each of the three color spaces and produced for each of seven regions of the retinal image in order to obtain the effects of the spatial variations on image quality.

The seven regions where the features were calculated are shown in Figure 3-6.

¹ CIELab is a color model developed by the Commission Internationale de L'Eclairage, that represents perceptual uniformity and meets the psychophysical need for a human observer. This model matches the sensitivity of human eyes with computer processing, whereas RGB color space didn't have such a property [78].

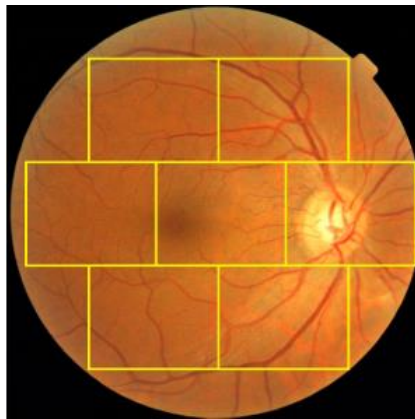


Figure 3-6 A yellow grid is placed showing the seven regions where the features are calculated (adapted from [13]).

The first set of features calculated, would provide a complete characterization of the luminance parameter in the image and all the features were calculated for each color channel:

- Mean intensities - where low or high exposure would be a reducing image quality factor;
- Skewness – a measure of symmetry, or the lack thereof;
- Kurtosis – a measure of whether the histogram of pixel intensities is peaked or flat relative to a normal distribution;

The second set of features would provide the characterization of the contrast parameter in the image:

- The variance of the intensities – within each of the seven regions in Figure 3-6 and each color channel, where low variance means low contrast, regardless of overall image brightness;
- Co-occurrence contrast from Haralick features – the relationship between a pixel and other pixels in its neighborhood;
- Entropy from Haralick texture – measures the quantized gray levels, where an even distribution (high contrast), will have the largest possible entropy value;
- Spatial frequency - is a measure that is affected by contrast and noise, where sharp edges, like those produced from retinal vessels, will increase spatial frequency.

With all the calculated features, the authors tried to find which of the characteristics were most relevant to the quality assessment, through their weights. The most important feature was the spatial frequency with a weight of 51.6% and in an overall classification, the first 15 features studied

contributed to 90.4%. Without discarding any feature, in a set of 200 non-indexable images and using two different fields of view (FOV) of 30° and 45°, they obtained 100% sensitivity and 96% specificity.

In 2014, *Veiga et al.*, [12] developed an approach based on the analysis of retinal quality parameters such as focus (sharpness) and field of view (FOV), in order to distinguish normal images from poor quality images.

The authors presented a method, never seen before in the literature, to analyze the quality of retinal fundus images where noise regions were extracted, and the focus was analyzed, using Wavelet-based, Chebyshev-based and statistically based measures. After these features were extracted, they were used as the input of a fuzzy inference system (FIS).

They proposed a quality system that would follow three blocks of processing. The first block was where they would calculate the FOV mask using the green channel extracted from the RGB retinal image (highest contrast channel) and calculated a noise mask. This last mask targeted regions of irregular illumination, that had very light or very dark areas. The optic disc was then detected to determine the most clinically relevant mask, which by the authors' analysis, was next to the macula. The binary mask of the FOV plus the noise mask is then analyzed to see if their common area was larger than a predefined threshold. This block ended with a classification phase, where the image was approved or not for a next stage (if this image did not have the minimum quality was not approved).

The last phase was the focus evaluation, where a classifier analyzed the input image.

The following figure, Figure 3-7, shows two examples of retinal images exhibiting slight decentralization at acquisition (a) and low pupil dilation (e). The images (b) and (f) are the FOV masks obtained by the mask FOV algorithm. Images (c) and (g) illustrates the light and dark noise masks obtained from the bright and dark mask algorithms. A logical OR is applied to obtain the final noise mask, as shown in (d) and (h).

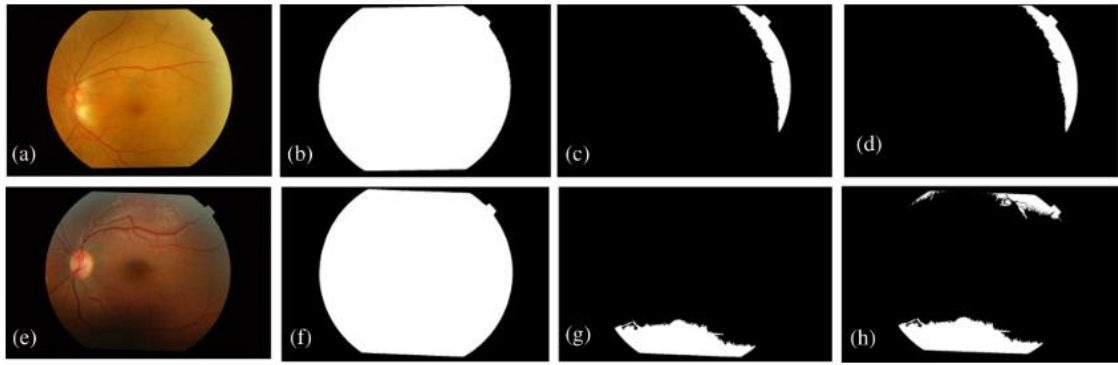


Figure 3-7 Digital retinal images with their FOV and noise mask (adapted from [12]).

To test this approach, *Veiga et al.*, used three datasets, MESSIDOR with 200 images (100 original and 100 artificially defocused), a dataset with real images, and a dataset with MESSIDOR + real images. The best results were achieved in the group of MESSIDOR artificially defocused images, obtaining an AUC=0.9946 and an classification accuracy of 98%.

Despite these good results, the authors concluded, that there are points of improvement in the speed of the algorithm and the use of more noise and real blurred images to resemble the real day-to-day problems of ophthalmologists.

3.2.2 DEEP LEARNING IQA APPROACHES

In the last years, almost all the classification methods used to classify fundus images depend on the type of features, that are based on generic quality parameters and are not generated generically in new datasets.

On the other hand, human experts relied on subjective capacities to identify poor quality images and had to be able to adapt to new scenarios based on new data. However, when the assessment is subjective, it only depends on the perception of what good quality is for the photographer. To overcome this fact, solutions have been developed that, although they require large amounts of data for their validation, reduce significantly the subjectivity and bias of existing algorithms.

These solutions start from Deep Learning, that solves this central problem in representation learning by introducing representations that are expressed in terms of other, simple representations [23]. Deep Learning allows the computer to build complex concepts out of simpler concepts, as presented in chapter 4.

Tennakoon et. al, [24] in 2016 presented a method for retinal IQA that uses Deep Learning computational algorithms to find if the images have sufficient quality or don't have sufficient quality for automated analysis – a binary classification approach. The authors used two different CNN architectures (model parameter estimation techniques).

The first network is a shallow CNN trained from scratch – with fewer parameters than conventional CNN and the second network is the AlexNet network, using transfer learning and pre-trained with the natural images from the ImageNet competition.

The shallow network consists of three convolutional layers with 96, 256 and 256 convolutional filters of kernel size 11 x 11, 5 x 5, 3 x 3. Each of the convolutional layers is followed by the ReLU activation functions and max-pooling layers (with kernel size 3 x 3).

Then the output of the feature maps of the previous convolution layers served as input for two fully connected layers. The last layer of the network was a softmax classification layer. In order to prevent overfitting, they used Dropout regularization in the last two fully connected layers and used the Batch Normalization to fill the problem called Internal Covariance Shift (where the distribution of the inputs of each layer changes with the training).

The authors trained the shallow network with cross-entropy loss function and this loss was optimized with stochastic gradient descent (SGD) with momentum=0.9, 30 epochs, weight decay=0.0005, learning rate=0.01 decreased by a factor of 10 every 10 epochs.

Figure 3.2.2-1 illustrates the shallow network used in fundus image quality assessment.

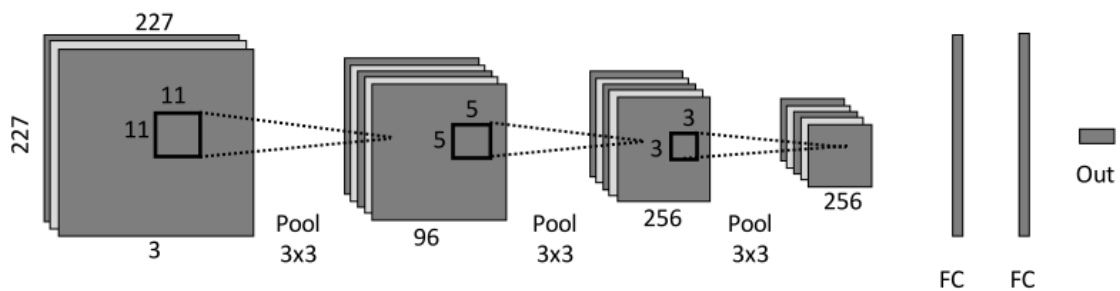


Figure 3-8 Architecture of the shallow network (adapted from [24]).

The second network is the AlexNet and is published in [25] consisting of five convolutional layers, three pooling layers, two local response normalization layers, and two fully connected layers.

Tennakoon et al., trained four separate classifiers, with 5-fold cross-validation:

- AlexNet-FT – a single layer Neural Network, that is fine-tuning the last layer of the original AlexNet;
- AlexNet-SVM – AlexNet with linear support vector machine;
- AlexNet-BT – AlexNet with boosted trees;
- AlexNet-KNN – AlexNet with k-Nearest Neighbours.

The dataset used contained 908 ungradable images and 944 gradable retinal images, all with 45° FOV and non-mydratic, with a resolution of 2812 x 2442 and the dataset was split randomly into 75% training and 25% test segments.

Data Augmentation was used to train the two networks with translations and rotations in a set of fixed angles (6° to 210° with resolution of 6°) to make the network rotation invariant.

The results of the classification using the test set can be seen in Table 3-1.

In table 3-1, the shallow network trained from scratch has achieved very high accuracy with 99.12% sensitivity – that indicates that this network was able to learn the necessary information for image quality classification, with only 3 convolutional layers, and the AlexNet with fine-tuning has higher sensitivity=99.55% but the same accuracy as shallowNet. However, the other classification models had lower accuracies, in the order of 96 to 97% (with the pre-trained weights).

These results have been concluded by the authors to be satisfactory, where classifiers that extract features using a pre-trained network can perform quite close to networks that are fully trained from scratch, whereas CNN's that are trained from scratch also has the advantage of being computationally less complex.

Table 3-1 Classification accuracy results on the test set for five classification models (adapted from [24]).

| Network | Accuracy (%) | Specificity (%) | Sensitivity (%) |
|---------------------|--------------|-----------------|-----------------|
| Proposed shallowNet | 98.27 | 97.46 | 99.12 |
| Alexnet-FT | 98.27 | 97.03 | 99.55 |
| Alexnet-SVM | 97.19 | 95.38 | 99.12 |
| Alexnet-BT | 96.98 | 99.15 | 94.71 |
| Alexnet-KNN | 96.98 | 96.19 | 97.80 |

In the same year, *Mahapatra*, [26] proposed a novel method based on combining unsupervised learning with local saliency maps and supervised learning with convolutional neural networks (CNN).

The authors were inspired by the *Itti et al.* article [27] where the original models for saliency maps can be found, but while their approach highlights the single region that is most salient and pixels outside the salient region have no importance, *Mahapatra* approach, proposes a local saliency map method that calculates the saliency value of each pixel incorporating both local and global features. The resized color image is converted to grayscale intensity, texture and curvature feature maps and the multiscale saliency maps are generated with these feature maps above.

It is possible to visualize the course of construction of the saliency map, in Figure 3-9 from the original image of good quality in (a), (b) - (d) represents the creation of the saliency maps by the way the state of the art approaches them [27], [28], [29] (e) - (g) construction of the local saliency maps by the method of the author *Mahapatra*. with different scales, (h) poor quality original image, (i) - (k) the respective salient maps for the image of poor quality by the authors method.

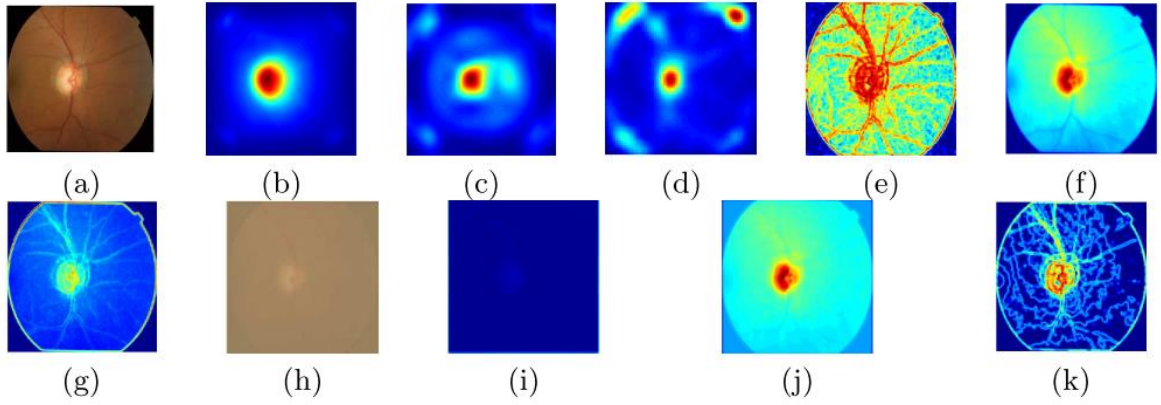


Figure 3-9 Local and global saliency maps obtained through the methods [5-7] and the authors' method *Mahapatra* (adapted from [26]).

The CNN in the proposed approach was fed with 512 x 512 input patches of retinal images and is composed of 5 convolutional layers followed by 2 x 2 max-pooling layers which downsampling the images to half the input dimensions to 256. The first convolutional layer had 10 kernels of 11 x 11 dimensions and the last layers (FC layers), had 4000, 2000 and 1000 nodes followed by a softmax classifier that outputs the class label as either gradable or ungradable (binary classification).

They used Stochastic Gradient Descent (SGD) optimization function and dropout with a probability of 0.5 in the second fully connected layer to speed the training time and half of the outputs of this layer are randomly masked. Figure 3-10 illustrated the architecture of the proposed CNN network and in Figure 3-11 the learned filters from the final convolutional layer.

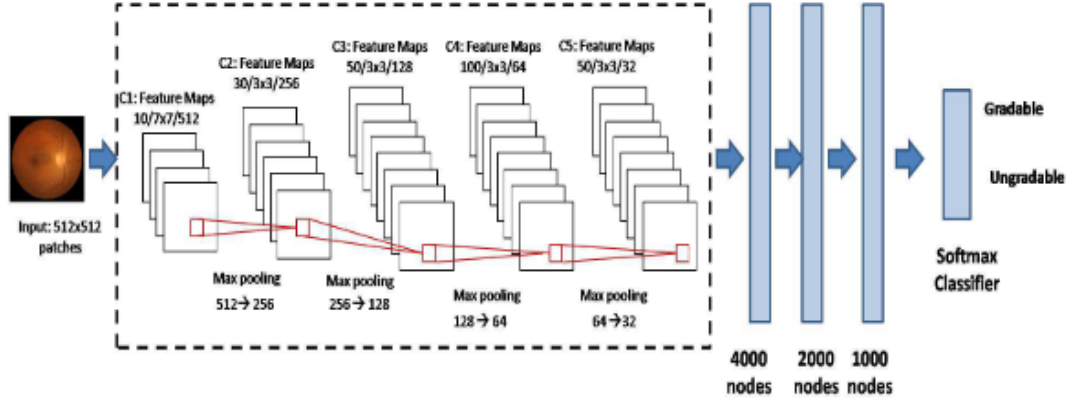


Figure 3-10 CNN architecture proposed in the present approach (adapted from [26]).

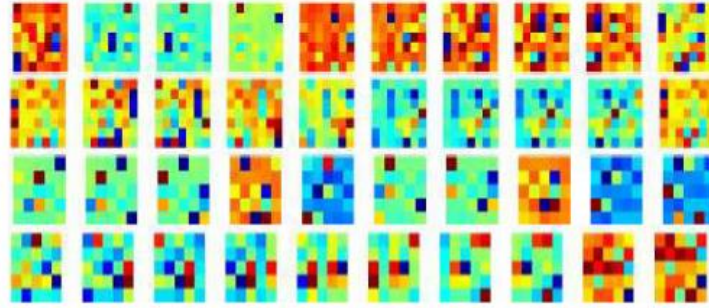


Figure 3-11 Learned filters from the last convolutional layer (adapted from [26]).

The dataset used for the study had 9653 ungradable retinal images and 11347 gradable images (more gradable than ungradable – little unbalanced), and all the images had 45° FOV and were non-mydratic. All the image intensities in the image were normalized between [0,1] and resized to 512 x 512, and there resized images were subject to data augmentation (flipping, rotation, translation, and contrast changes).

The image vector saliency maps (f_1) and the 1000 dimensional feature vector from the last FC layer of the CNN (f_2) were used to train two different Random Forest (RF classifiers denoted RF_1 – supervised image features and RF_2 - unsupervised image features).

The results are in Table 3.2 and the metrics studied were sensitivity (correctly identified gradable images), specificity (correctly identified ungradable images), accuracy and p-value. RF_{1+2} stands for *Mahapatra* method; RF_{All} is the method where the feature vectors f_1 and f_2 are concatenated to train the Random Forest; SVM_{All} is the support vector machines using f_1 and f_2 ; $RF_{1+2} + SM$ is the weighted combination of outputs of RF_1 and the softmax classifier from the CNN. The authors

obtained high accuracy of 97.9%, also higher sensitivity and specificity, than Dias [30], and Niemeijer [31], and significantly better than Paulus [1].

Table 3-2 Classification results for different methods compared to CNN (adapted from [26]).

| | RF_{1+2} | RF_{All} | SVM_{All} | Paulus | Dias | Niemeijer | $RF_1 + SM$ | RF_1 | SM |
|------------|------------|------------|-------------|--------|------|-----------|-------------|--------|------|
| <i>Sen</i> | 98.2 | 95.4 | 95.1 | 94 | 96.1 | 96.7 | 97.9 | 92.2 | 93.4 |
| <i>Spe</i> | 97.8 | 94.6 | 94.2 | 90.1 | 95.4 | 96.0 | 97.8 | 91.8 | 92.4 |
| <i>Acc</i> | 97.9 | 94.7 | 94.5 | 91.4 | 95.6 | 96.2 | 97.9 | 91.9 | 92.8 |

As final notes, the authors conclude that the good results compared to those of the state of the art are due to the fact that the models were tested with a larger test dataset, having combined two different sources of information - unsupervised information from visual saliency maps and supervised information from trained CNN, and finally, the computation times are low (4.7 seconds for classification) which allows a quick assessment of retinal image quality.

4 DEEP NEURAL NETWORKS PRINCIPLES AND FUNDAMENTALS

Computer vision (CV) is a process and a branch of computer science, that involves capturing, processing and analyzing real-world images and video, allowing machines to extract meaningful and contextual information from the physical world. Today, Computer Vision is the key means and a foundation of testing and exploiting deep learning models that are propelling the evolution of artificial intelligence toward useful and practical applications.

A significant part of AI field deals with planning for system/machine which can perform mechanical actions. This type of processing needs a high amount of training data (input data), provided by a computer vision system algorithm, acting as a vision sensor and give away high-level of information about the environment and the machine.

This chapter presents the key concepts for building a Deep Learning model as well as presenting more general knowledge so that someone new to the area can understand them.

4.1 MACHINE LEARNING AND DEEP LEARNING

Machine learning (ML) is a category of artificial intelligence algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. As availability of computational capacity and data has increased, machine learning has become more and more practical over the years, to the point of being almost ubiquitous.

Most machine learning algorithms can be divided into the categories of supervised learning and unsupervised learning.

Unsupervised learning algorithms experience the dataset containing many learnable features and to extract useful properties from them. An unsupervised learning algorithms example is clustering, which consists of dividing the dataset into clusters of similar examples among the variables present in the data, having no way to know if they were correctly grouped.

Supervised learning problems are categorized as regression and classification, and the approach of learning is different from the unsupervised algorithms, since the datasets for learning and training of the algorithms contain, besides the data to be evaluated, also the label associated to each input data, as the example of the study of Iris dataset. This type of learning involves observing several examples of training images and associating them with a predictive value [23].

The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs—not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called generalization. Typically, when training a machine learning model, the training dataset is known, and to know how a model is performing with this seen data, some error measures can be computed, called the training error. This is called an optimization problem.

To have a successfully machine learning algorithm, not only the training has to have low loss, but also the unseen data (validation set and test set). The generalization error is defined as the expected value of the error on a new input.

The generalization error of a machine learning model is usually measured by its performance on a test set of examples that were collected separately from the training set. Therefore, the factors that most contribute to a good algorithm performance, are its ability to make the training error small and the gap between the two curves of learning (train and test curves) has to be small.

When a model does not learn and is not able to obtain a low error value in the training set, it is occurring the phenomenon called underfitting. When the gap between training error and test is too large, that is overfitting, which has been one of the major problems faced in Machine Learning. In Figure 4-1 it can be seen the overfitting occurrence.

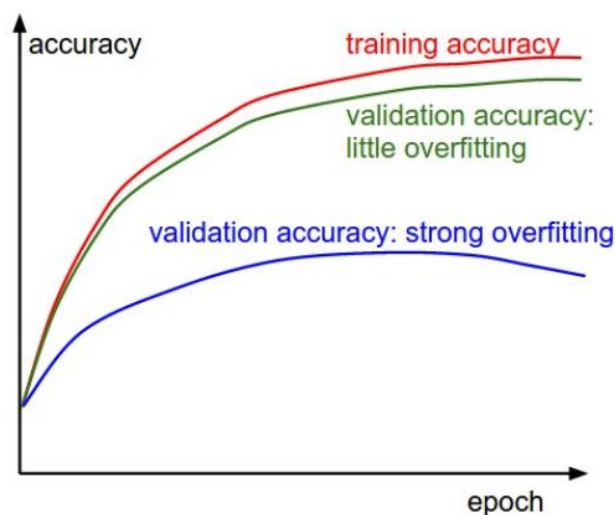


Figure 4-1 Learning curves with and without overfitting. Adapted from [32].

Deep Learning is a subset of a more general field of artificial intelligence called machine learning, which is predicated on this idea of learning from example. In Deep Learning, instead of teaching a computer a massive list rules to solve the problem, it is given a model with which it can evaluate examples and a small set of instructions to modify the model when it makes a mistake [33].

It consists of a pipeline of convolutions and subsampling operations, applied at various scaled versions of the original image, to handle faces of different sizes. This pipeline performs automatic feature extraction and classification of the extracted features, in a single integrated scheme. The full process is implemented via a convolutional neural network architecture, which offers the advantage of being trained to automatically derive all parameters, governing feature extraction and classification. Figure 4-2 shows the comparison between Machine Learning and Deep Learning and in Figure 4-3, the progress that all areas of artificial intelligence have been taking over the past 60 years.

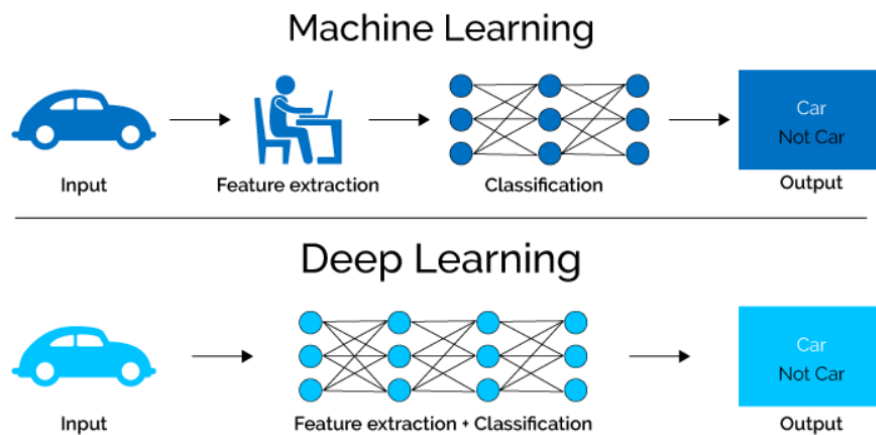


Figure 4-2 Comparison between Machine Learning and Deep Learning. Adapted from [34]).

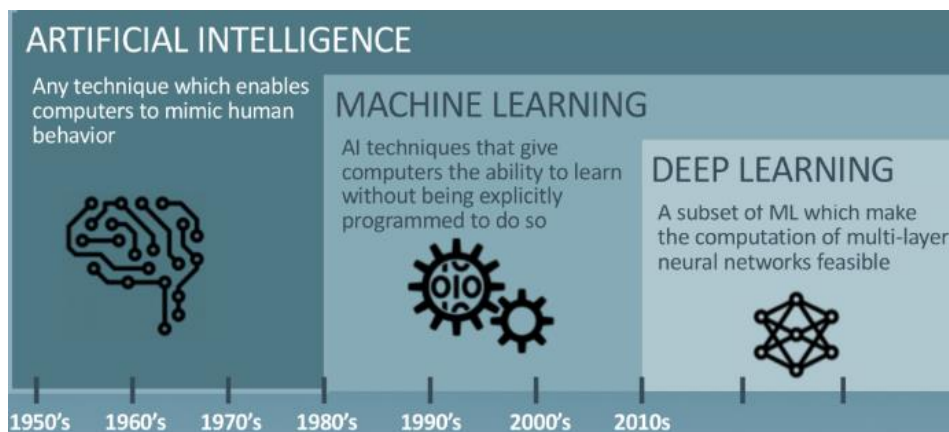


Figure 4-3 Progress chronology of Artificial Intelligence, Machine Learning and Deep learning concepts. Adapted from [35]).

4.1.1 NEURAL NETWORK

Neural network is a generic term in Deep Learning that works on the basis of the structure and functions of a human brain. Like the human brain has interconnected neurons that constantly transmit signals, a neural network also has interconnected artificial neurons that transmit data among each other and are called as nodes. These neural networks are called as Artificial neural networks (ANNs) [36].

Artificial neural networks (ANNs) model the relationship of learning units, called neurons or perceptrons, that convert input signals (e.g. picture of a dog) into corresponding output signals (e.g. the label “dog”), forming the basis of automated recognition. Taking the example of automatic recognition, the process of determining whether a picture contains a dog involves an activation function. If the picture resembles prior dog images the neurons have seen before, the label “dog” would be activated. Hence, the more labelled images the neurons are exposed to, the better it learns how to recognize other unlabeled images. This process is called training.

A traditional neural network, also called Multi-layer perceptron (MLP), consists of 3 types of layers: input layers, hidden layers and output layers, taking x_1, x_2, \dots, x_n number of inputs, each of which is multiplied by a specific weight w_1, w_2, \dots, w_n and added to a bias b_1, b_2, \dots, b_n . These weighted inputs and biases are summed together producing the logit $z = b + \sum_{i=1}^n w_i \cdot x_i$. The logit is then passed through a function f to produce the output $y = f(z)$ or $y = f(x \cdot w + b)$. In other words, the output computation is done by the dot product of the input and weights, adding the bias term to all the layers in the network [33]. The representation of a single neuron in an artificial neural network, is presented in Figure 4-4, and Figure 4.5 [37] represents an artificial neural network.

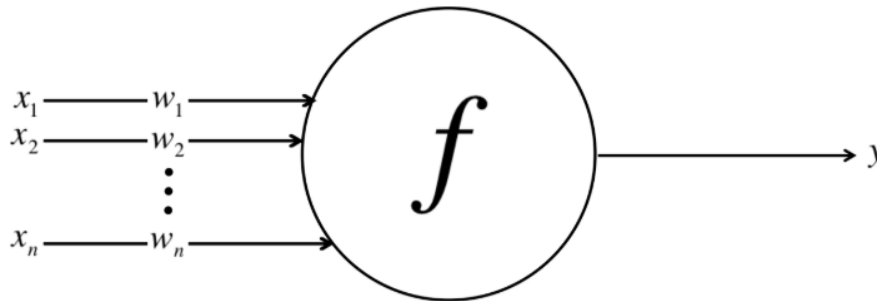


Figure 4-4 Representation of a neuron in an artificial neural network Adapted from [33].

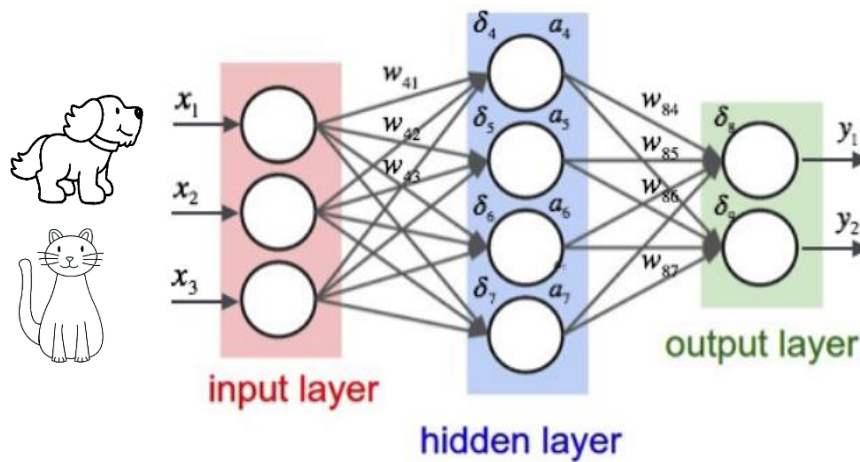


Figure 4-5 Example of an Artificial neural network, with three layers.

The first layer (input layer) receives the input data, such as pictures of a dog and cat. The middle layer(s) of neurons, called hidden layers, transform the inputs, doing several calculations and feature extractions, according to the weights and bias from the last layer. Finally, the last layer, called output layer, computes the final answer and predictions, obtained through network learning. In this particular case, the network has two output nodes y_1 and y_2 , having a response equal to $[0,1]$ if the image contains the label “dog”, $[1,0]$, if the image contains the label “cat”, and $[0,0]$, if the network doesn’t predict any of the classes.

The process by which the network learns is based on feed-forward propagation, where the features are input to the network and fed through the following layers to produce the output activation, where for example, in the hidden layer, the activation obtained in a specific neuron is the combination of the weights and biases of the input layer and the weighted combination of all the input values.

4.1.2 LOSS FUNCTIONS

Each loss function is used depending on the type of problem to be solved. In the case of regression problems, mean squared error (MSE) is used, and in classification problems, logarithmic losses, log losses or cross-entropy losses like binary cross entropy and categorical cross entropy, are used. The cross-entropy losses measure the performance of a classification model whose output is a probability value between 0 and 1, and the loss increases as the predicted probability diverges from the actual label. A perfect model would have a log loss of 0.

These loss functions are derived from Maximum likelihood principle, where given Θ parameters of the model and a model generated with D inputs and P predictions, the idea of this principle is to find the Θ that maximizes the predictions done by the model, $P(D | \Theta)$. In binary classification, the loss function that maximizes the predictions in the model is the binary cross entropy and in multi-class classification is the categorical cross entropy. Figure 4-6 shows a graph with the range of possible loss values given a true observation, for example, when the neural network true predicts the label “dog”. As the predicted probability approaches 1, log loss slowly decreases.

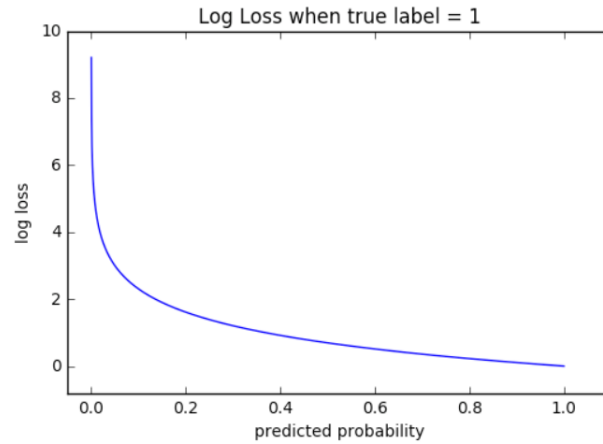


Figure 4-6 Cross-entropy (log loss) when the true label is 1 and the predicted label is 1 too.

4.1.3 ACTIVATION FUNCTIONS

In order to learn complex relationships in neural networks, neurons that employ some nonlinearities, also called activation functions, are used. These functions are the mechanisms by which an artificial neuron processes information and passes it throughout the network [36]. In Figure 4-7, it is presented a neuron, where $a_1, a_2, a_3, \dots, a_n$ are the activation functions, $w_1, w_2, w_3, \dots, w_n$ are the weights, b is the bias, z is the logit $z = b + \sum_{i=1}^n w_i \cdot x_i$ and σ is the function calculated in the neuron.

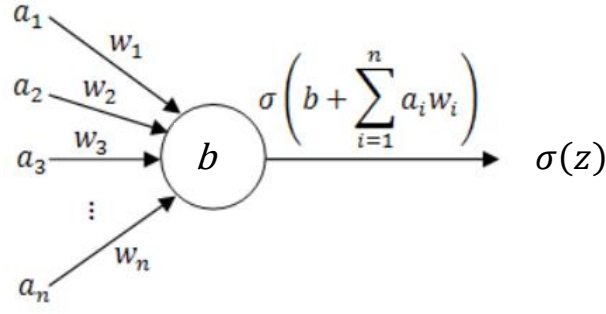


Figure 4-7 A perceptron representation with the weights, activation functions, the bias and the function $\sigma(z)$.

There are some types of activation function: sigmoid, tanh, ReLU and softmax. Sigmoid neuron, which uses the function (4.1), takes a real valued number for logit z , that is in a range between 0 and 1. The value is very close to 0, when the logit is very small and close to 1 when the logit is very large [33]. Sigmoid is a popular choice, which makes calculating derivatives easy and is easy to interpret [36].

Sigmoid units can be used in the output layer in conjunction with binary cross entropy for binary classifications [38].

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.1)$$

Tanh neuron, which uses the function (4.2), is a s-shaped nonlinearity, ranges between -1 and 1 and the output is zero-centered. When this kind of nonlinearity is used, the tanh neuron is often preferred over the sigmoid, because tanh is zero-centered [33], [36].

$$\sigma(z) = \tanh(z) = 2\sigma(2x) - 1 \quad (4.2)$$

Rectified linear unit (ReLU) applies the function (4.3) to all of the values in the input volume. In basic terms, this layer just changes all the negative activations to 0. This unit is more commonly used as a hidden unit in the recent times, because results show that ReLU lead to large and consistent gradients, which helps gradient-based learning and better convergence [36], [38].

$$\sigma(z) = \max(0, z) \quad (4.3)$$

To output a vector with probability distribution in the output layer, over a set of mutually exclusive labels and to know how confident the predictions are, it's commonly used the softmax activation function [32]. The output of this neuron also depends on the outputs of all the other neurons in its layers, because it's required the sum of all the outputs to be equal to 1.

A strong prediction would have a single entry vector close to 1, while the other prediction values were close to 0. A weak prediction has all the possible labels more or less equally like.

This kind of layer is typically used as an output layer for multi-classification tasks in conjunction with the cross-entropy loss function. Figure 4-8 represents a softmax layer and Figure 4-9 represents the different types of activation functions used in neural networks.

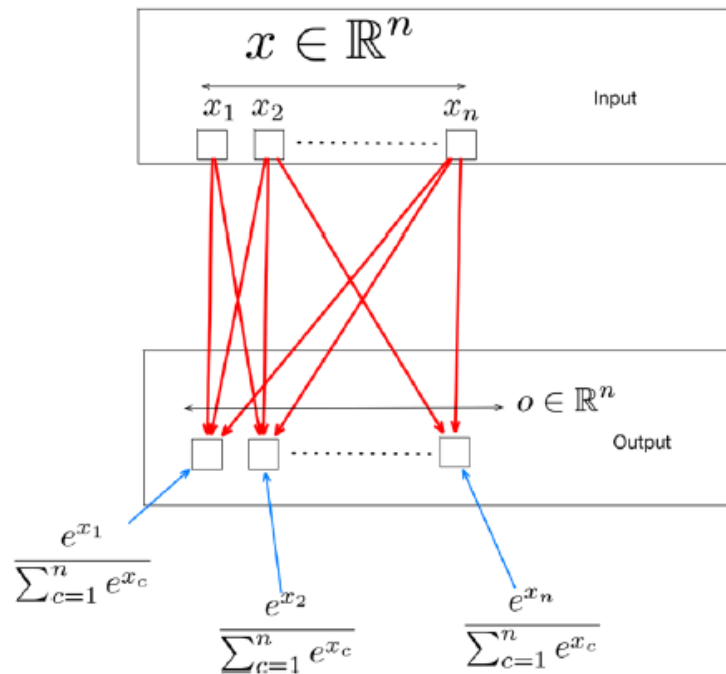


Figure 4-8 Softmax layer. Adapted from [38].

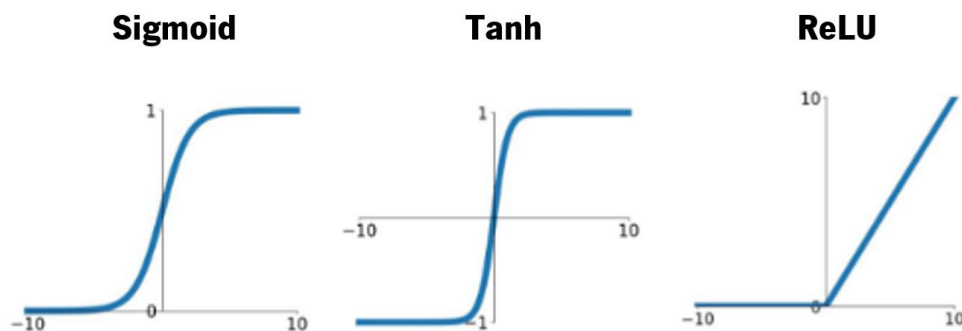


Figure 4-9 Neural network activation functions.

4.1.4 GRADIENT DESCENT, LEARNING RATE AND OPTIMIZATION FUNCTIONS

In order to minimize the loss or penalty in neural networks, it's necessary to use a strategy that tackle the training process, by finding the right values of weights, bias and the minimum error. This strategy is called gradient descent and is the algorithm used to train each of the individual neurons in the neural networks.

In addition to the weight and bias parameters, learning algorithms also require some additional parameters, called hyperparameters that carry out the learning process, like the learning rate.

Learning rate (LR) is the hyperparameter responsible for the directions and steps needed to convergence and proximity of the local minimum, given by the gradient descent algorithm.

To understand how the error J minimizes, a 3 dimensional-space with circular contours is shown in Figure 4-10. The black lines are the steps that move perpendicular to the contour and show how close the minimum is. Picking a good learning rate is a hard problem, because when it's too small, the training process risks taking too long, but if it's too big, it's mostly likely to start diverging away from the minimum error [33].

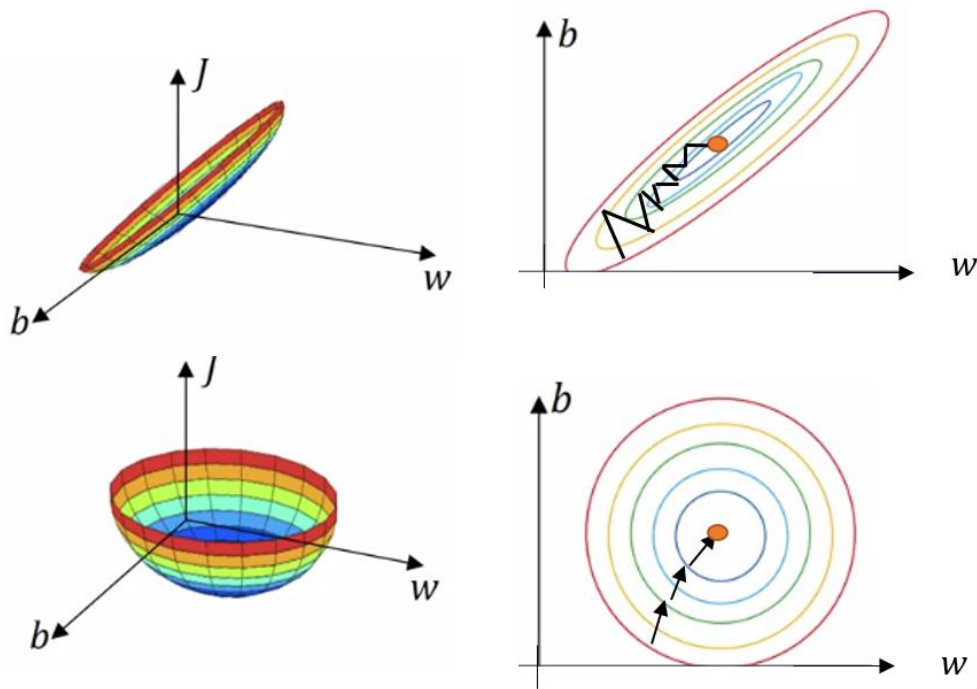


Figure 4-10 Gradient Descent Algorithm 3D visualization. Adapted from [39].

When gradient descent approaches a minimum, a bad learning rate can cause it to oscillate between around the minima. To overcome this problem and select the best learning rate, there are some optimization functions.

Stochastic Gradient Descent (SGD) or incremental gradient descent is an iterative method of neural network optimization. It's called stochastic because samples of images are selected randomly (shuffled) instead of appearing in the order they appear in the training set. As the algorithm sweeps through the training set, it performs the above update for each training example. Several passes can be made over the training set until the algorithm converges.

While stochastic gradient descent remains a popular optimization strategy, learning with it can be slow. To accelerate the learning process, the momentum method can be implemented, especially in the face of high curvature, small but consistent gradients, or noisy gradients [23].

Adam (Adaptive Moment Estimation), RMSprop (Root Mean Square Propagation) and Adagrad are other kind of optimization algorithms that are often used in neural networks and use adapted learning rates.

Adam is an extension to SGD that has recently seen broader adoption for machine learning applications. This algorithm has become popular, getting good and fast results [40].

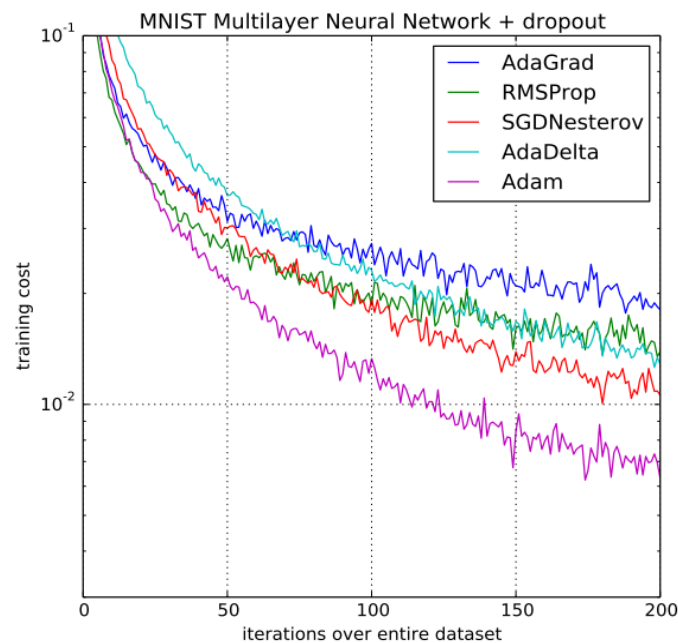


Figure 4-11 Loss curves of the different optimization functions. Adam is the optimization function with better learning, since the loss curve decreases continuously. Adapted from [40].

4.2 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN), introduced in 2010, by LeCun et. al. [41] are specialized and bioinspired hierarchical multilayered neural networks, that have a known grid-like topology, like a time series (1-dimension grid) or an image (2-dimension grid).

The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution, that is a matrix multiplication [23].

These neural networks, have gained wide popularity in computer vision, and their success is mainly attributed to faster processing (GPU), the use of non-linearity functions such as rectified linear units (ReLU) and dropout regularization [42].

These networks have been tremendously successful in practical applications, generally used for image detection and classification tasks, such as face recognition [43], handwriting recognition [41] and sentence classification [44].

4.3.1 CONVOLUTION OPERATION AND FEATURE MAPS

CNNs combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and spatial subsampling [45]. These concepts will be discussed throughout the chapter.

The first step layer in a CNN is the convolution layer, that consists in a set of learnable filter (also called kernel), that slide over the input images to compute the convolution operation. Every filter is normally spatially small but extends through the full depth of the input volume. A typical filter of the first convolution layer might have receptive field (filter size) of $5 \times 5 \times 3$ (i.e. 5 pixels width, 5 pixels height and depth 3, because an RGB image has 3 channels. In case the input images are grayscale, the depth/number of channels is equal to 1. During the forward pass, each filter slides across the width and height of the input volume and computes dot products between the entries of the filter and the input at any position. As the filter slides, the width and height of the input volume produces a 2-dimensional feature map. Intuitively, the network will learn filters that activate when some type of visual feature, such as an edge or pattern, is present. An entire set of filters in each convolution layer (e.g. 12 filters), will produce a separate 2-dimensional feature [33], as can be seen, in Figure 4-12 [25]. In figure 4-13 [46] is represented the convolution operation between the input image as I with kernel as K .

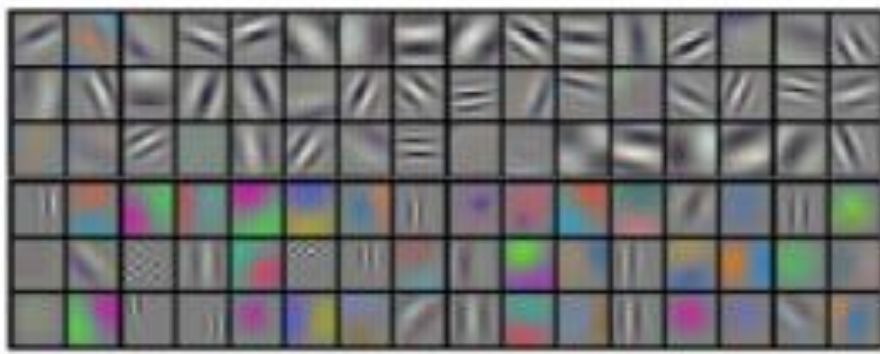


Figure 4-12 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images.

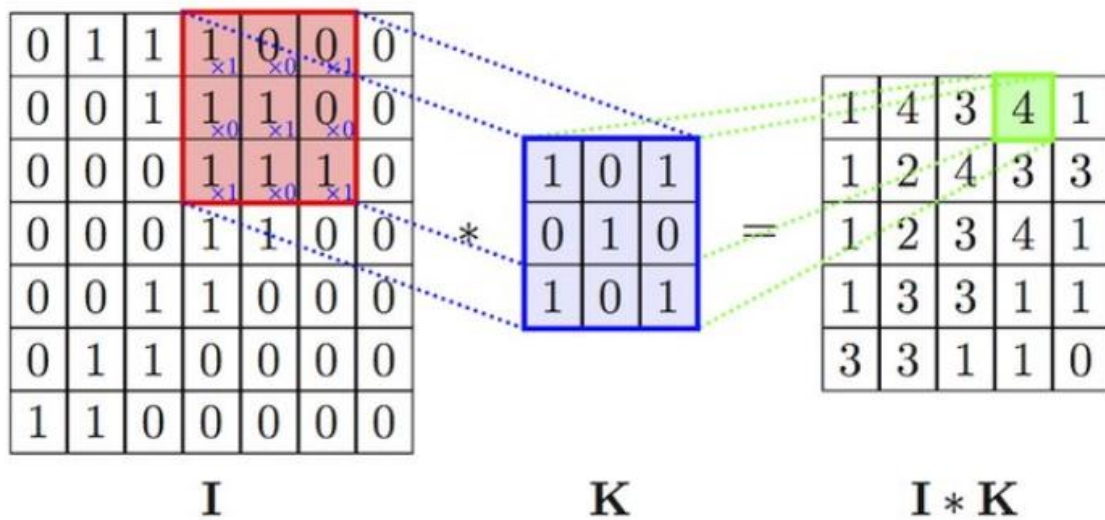


Figure 4-13 Convolution operation of input image I with kernel K , resulting $I * K$.

4.3.2 MAX POOLING, STRIDE AND PADDING

It is common to insert a pooling layer in-between successive convolutional layers in a CNNs.

These type of layers have the power to reduce dimensionally, downsampling the feature maps and sharpen the located features [33]. Reducing the dimension of the features also corresponds to reduce the number of parameters and computation in the network, helping to control overfitting.

The pooling layer operates independently over each activation map (ReLU activation map) with, for example, a 2x2 dimension filters and stride 2, downscaling the input dimensions by half. Figure 4-14 shows an example of a max pooling operation, where in each of the colored squares, the max value is taken, and the depth dimension remains the same [47].

Stride is a parameter that specifies how many pixels the filter skip over the input feature map. When the stride is 1, the filters move one pixel at a time. When the stride is 2, the filters jump 2 pixels at a time. This will produce smaller output volumes spatially. Another parameter, frequently and convenient to use in convolutions is the parameter padding, which means, that the input volume is padded with, for example, zeros around the borders, which is called zero-padding. The size of this zero-padding is a hyperparameter and allows to control the spatial size of the output volumes [47], as seen in Figure 4-15 [48].

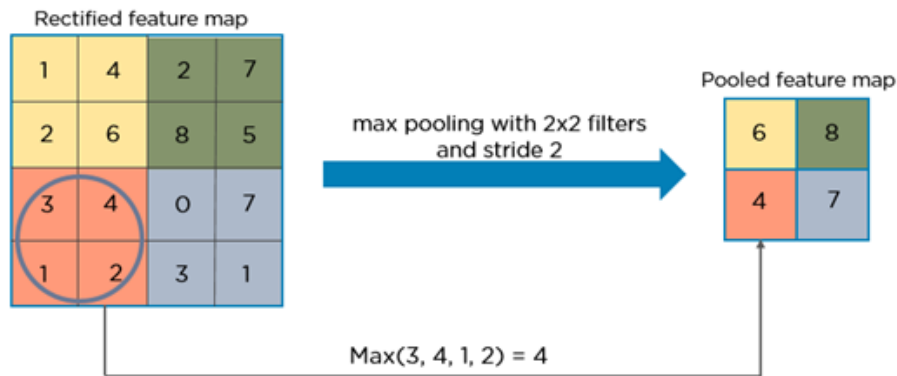


Figure 4-14 Max pooling operation. Adapted from [49].

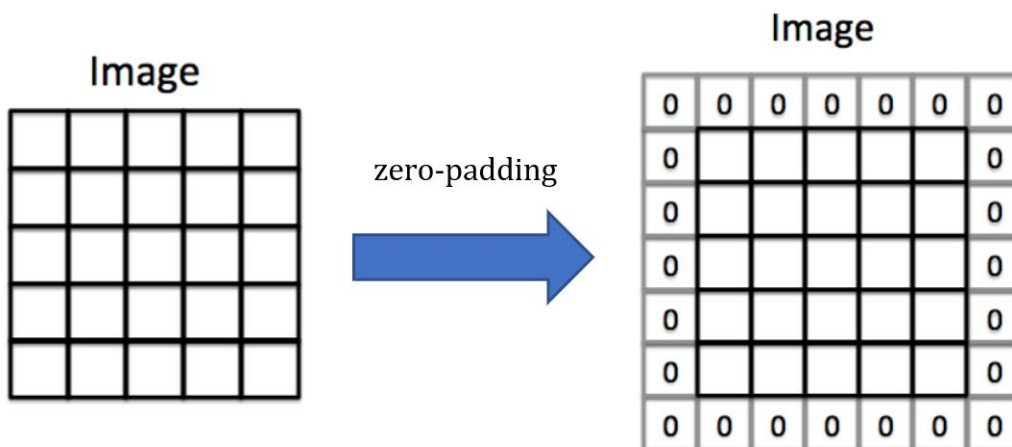


Figure 4-15 Padding operation in an input image.

4.3.3 WEIGHTS INITIALIZATION

In Convolutional neural networks, weights are usually initialized randomly, having a bad impact in the training session. This kind of initialization takes significant amount of repetitions to converge to the least loss and reach to the ideal weight matrix. The problem is, this kind of initialization is prone to vanishing or exploding gradient problems.

One way to reduce this issue is carefully choosing the weight initialization, that is a parameter that is included in the fully connected layers.

There are some weight initializations available:

- random uniform weights initialization that generates tensors with uniform distribution;
 - random normal weights initialization that generates tensors with normal distribution;
- Xavier's random weights initialization [50] that samples from truncated normal distribution centered in 0, represented in equation:

$$std\ dev = \sqrt{\frac{2}{fan_in + fan_out}} \quad (4.1)$$

where fan_in is the number of input units in the weight tensor and fan_out is the number of output units in the weight tensor [51].

4.3.4 NORMALIZATION

4.3.4.1 LINEAR NORMALIZATION

If the input data are on very different scales and the range of values is significant, then w (weights) parameters may assume very different values and the most likely would be if the cost function were to look like the shape of Figure 4-10 (section 4.2.4) (upper left and right), resulting in an elongated cost figure, where the probability of having the local minimum in the function is small. When normalizing the data, the cost function seems more symmetrical as shown in figure 4-10 (down left and right).

When executing the gradient descent algorithm over the cost function with non-normalized data, it is necessary to use a very low learning rate, since several steps and swings (usually small) between

the circular lines (Fig 4-10 upper right) may be needed to reach a local minimum. However, using the gradient descent in the more spherical contour, it can go directly to the minimum, with larger steps, whatever the starting point. Therefore, the cost function is easier and faster to optimize when data is on similar scales, for example between $[0,1]$ or $[-1,1]$.

Linear normalization transforms the pixels range $[0,255]$ to the value range of $[0,1]$.

By normalizing the input distribution, a better model can be achieved, which learns faster, converges to low error and doesn't get stuck at local minima, as Andrew Ng said in a lecture called Normalizing Inputs [39].

4.3.4.2 BATCH NORMALIZATION

Training convolutional neural networks, where the distribution of each layer's inputs changes during training, turns all the process complicated. This phenomenon is called internal covariate shift, that slows down the training, requiring lower learning rates, careful parameter initialization and a normalization of the layers inputs. To overcome this phenomenon, Batch Normalization can be implemented [52].

This kind of layer computes the mean and standard deviation of all the features, by shifting inputs to zero-mean and unit variance, making the inputs of each trainable layers comparable across features [23].

In addition to this function, Batch Normalization has proved to be a good method of regularization, making it possible, in some cases, to eliminate the dropout layer, reduce L_2 weight decay regularization and enable the use of higher learning rates, making the training session faster [52].

4.3.5 MODEL OPTIMIZATION AND REGULARIZATION

Methods of combating the overfitting phenomenon and variance reduction are called by regularization. Regularization usually modifies the objective function, minimizing it by adding extra parameters and penalizing large weights.

The batch normalization proved to be a good regularizer, as mentioned in the previous section and among this type of regularization, there are others that can penalize the weights or penalize neurons used during training.

The most common type of regularization is L_2 regularization, also known as weight decay regularization. This regularization technique decays or shrinks the weights found during learning and model optimization, favoring diffuse weights, vector relative to peaky weights vectors, avoiding that the weights become too big, throughout the training [33].

Another form of regularization is the max norm constraint which also has the function of restricting weights and preventing them from becoming large. This is imposed by a fixed value c that is normally between 3 and 4. One of the interesting properties of this regularizer is that the parameter vector does not go out of control, since, updates to the weights are always bounded.

The Dropout has the function of keeping only a neuron active with some probability p (p is a hyperparameter), during training, preventing the network from becoming too dependent of any small combination of neurons and introduces random noise to the training samples [33], [23].

Researchers such as [24] and [26] have used dropout between the last two fully connected layers and the value of $p = 0.50$ and have proved to be a good strategy of regularization. Figure 4-16 shows the representation of a standard network without dropout (a) and (b) corresponds to a network after the dropout is applied.

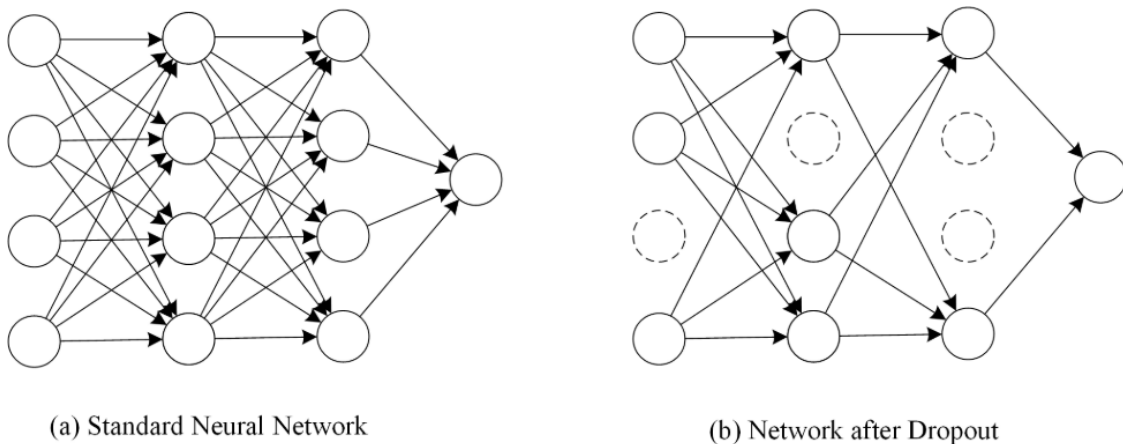


Figure 4-16 Dropout regularization. Adapted from [53].

5 RETINAL QUALITY ASSESSMENT EXPERIMENTS

RETINAL QUALITY ASSESSMENT EXPERIMENTS

The proposed retinal image quality assessment method using Deep Learning includes six main stages. The first step called *Data Acquisition*, involves searching for appropriate datasets to study the parameters that define quality (the parameters are described in subsections 3.1.1 to 3.1.5). The description of the datasets obtained, the type of images used, their size and the type of diseases/lesions that are acquired through the images are described in subsection 5.1 of this chapter.

The next step is called *Data Preprocessing*, where some image processing techniques are contemplated, for example, the application of a threshold to obtain binary masks or application of black bounding boxes to the uniformization of the dimensions of the images.

After processing the images, a file preparation follows, from loading pre-processed images to CSV files and then dividing the images into training, validation and test subsets (*Data Preparation*).

Then, these images are loaded to the CNN network, with network creation, compiling and training the networks. The last step is called *Predictions*, where model evaluations and values of classification metrics are obtained.

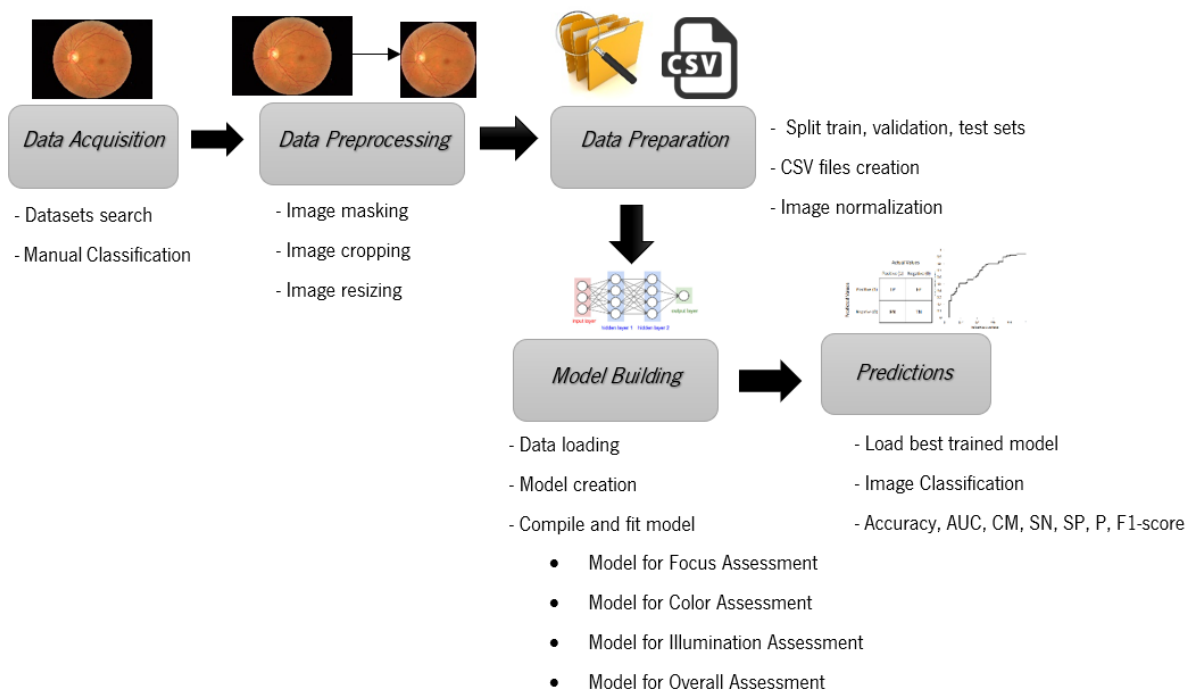


Figure 5-1 Methodology pipeline applied to automated retinal image quality assessment using a Deep Learning approach.

5.1 MATERIALS

In order to carry out the present work, different types of images and databases were used in order to cover different types of cameras and demographic diversity.

The images chosen were centered on the macula and centered on the optic disc, and these images first underwent a manual classification where the field definition (one of the image quality parameters discussed in subsection 3.1.1 of the present work) was analyzed. All images of the fundus of the eye were manually classified by two specialists and this classification was intended to cover all the parameters described in subsections 3.1.1 to 3.1.5.

The experimental setup used for the present work to preprocess the images, train the neural networks and to predict the image quality assessment was implemented in JupyterLab. JupyterLab is a web-based user interface for Project Jupyter, enabling to work with documents such as Jupyter notebooks, terminals, text editors in an integrated and extensible manner [54].

JupyterLab was runned with Python 3.5, to preprocess the images it was used OpenCV 2 and the experiments computer had the specifications presented in table 5-1.

Table 5-1 Specifications of the experiments computer.

| | |
|--------------------------|--------------------------------|
| Operated System | Ubuntu 16.04 LTS 64-bit |
| GPU | NVIDIA Quadro P6000 |
| GPU Memory | 24 GB GDDR5X |
| NVIDIA CUDA cores | 2340 |
| CPU | Intel Xeon, 12 cores, 2.70 GHz |

5.1.1 PROPRIETARY DATASET

The proprietary dataset used in the study was provided by APDP. It contains in its total 983 digital fundus photographs, with the acquisition in 45° FOV. The brand of the camera of photos acquisition is Canon and the model varies between CR-1 and CR1Mark2. There are four different image dimensions in the dataset such as: 3888 x 2592, 2812 x 1880, 3456 x 2299 and 2376 x 1580 pixels. The images are from diabetic retinopathy screening program, with images from R0 - without retinopathy to R2 - severe non-proliferative diabetic retinopathy (severe NPDR). Figure 5-2 contains examples of images that are available in the APDP dataset (with the preprocessing step described

in the subsection 5.2) where the image (a) contains R1 - Mild NPDR and is classified as macula-centered, with good field definition, with even illumination, well focused, with no visible artifacts and with normal clarity; the image (b) is also R1 – Mild NPDR, macula-centered, with good field definition, with even illumination, well focused, with artifacts, and with normal clarity; the image in (c) is R2 - Moderate NPDR, centered on the optic disc, with good field definition, with uneven illumination where the area around the macula is dark, contains artifacts and with normal clarity.

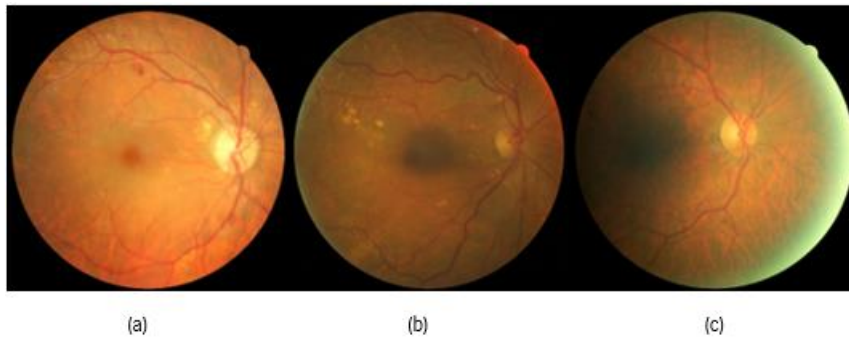


Figure 5-2 Examples of images present in the APDP dataset.

5.1.2 PUBLIC DATASETS

- IDRiD [55] – This dataset is known by Indian Diabetic Retinopathy Image Dataset, with 516 images, was created through an initiative called “Diabetic Retinopathy: Segmentation and Grading Challenge” at IEEE International Symposium on Biomedical Imaging (ISBI-2018), to detect and grade diabetic retinopathy and diabetic macular edema using retinal fundus images. No information is given about retinal image FOV and all of the images have a resolution of 4288 x 2848 pixels and are macula-centered.

Figure 5-3 contains examples of images that are available in the IDRiD dataset (with the preprocessing step described in the subsection 5.2) where the image (a) contains R2 - Moderate NPDR, with Risk of Macular Edema=2, and is classified as macula-centered, with good field definition, with even illumination, well focused, with no visible artifacts and with normal clarity; the image (b) is R1 – Mild NPDR, macula-centered, with good field definition, with even illumination, with a fair focus, with artifacts, and with normal clarity; the image in (c) is R0 – without DR, centered on the macula, with good field definition, with even illumination, without artifacts, and with normal clarity.



Figure 5-3 Examples of retinal images from the IDRiD dataset (adapted from [55]).

- EyePACS [56] – This dataset contains images of different models and types of cameras, which can affect the visual appearance of retinal left and right sides. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Some images have in the field definition and the acquisition was made with 30°, 45° and 60° FOV. The images present in these datasets are intended to detect lesions of diabetic retinopathy or if there are signs that point to the onset of this disease and to associate a degree between 0 - without retinopathy and 4 - proliferative retinopathy. There are images of varied quality, having images of good quality, of acceptable quality or without quality.
- STARE [57] – STARE stands for Structured Analysis of the Retina project, initiated in 1975. Contains a full set of 400 images, with expert annotations and manual classification done by experts.
- ROC [58] – ROC is the acronym for Retinopathy Online Challenge and was created in the University of Iowa. This challenge aims to help patients with diabetes and detect diabetic retinopathy.
- HRF [59] – HRF is High-Resolution Fundus image dataset, with 18 image pairs of the same eye from 18 human subjects using a Canon CR-1 fundus camera and a field of view of 45° and different acquisition setting. For each pair, the first image has poor quality and thus the examination had to be repeated.

5.2 METHODS

5.2.1 DATA PREPROCESSING

Before the quality analysis of retinal images, preprocessing operations are performed for a given image. The digital fundus photographs of the retina, consisting of different sizes and taken from different cameras, are used as input data of the preprocessing step.

To preprocess the images, it was used the OpenCV computer vision library [60]. This library was designed for computational efficiency and with a strong focus on real-time application, providing a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications [61].

The preprocessing step includes image masking, image cropping and image resizing.

The principle of image masking is based on labeling corresponding pixels of both retinal foreground and background throughout the image. The generated masks then follow the image cropping step, that is removing irrelevant image information such as useless black borders around the circular retina.

Finally, the process of resizing the images is followed, because at the end of the last preprocessing steps, the images contain all different heights and widths and to be loaded into the neural network, they must be of the same size standard.

The processes described above are presented sequentially in Figure 5-4.



Figure 5-4 Implemented preprocessing pipeline.

5.2.1.1 IMAGE MASKING

A retinal fundus photography consists of a colored circular region in the foreground (which is also the Region of Interest – ROI) on a black background and the process of labeling image pixels as the background is known as image masking.

Image masking allows distinguishing background from foreground through the use of a simple threshold. The use of simple thresholds [62], [63], and region growing [64] can be seen in the algorithms implemented in the literature.

To create the image mask, the following steps were followed (Appendix A-1):

1. The RGB image is converted to grayscale;
2. An empirically determined threshold is applied, with $t=10$, to the grayscale image;
3. The noise of the image results in 2 is reduced by applying a morphological operation method called Structuring Element with a 3×3 kernel size; then is applied an opening, that removes noises in the background and in the boundary of the retina [3];
4. A closing operation with a 3×3 kernel size is applied in order to close small holes or black points inside the foreground [65]. The kernel size was empirically determined.
5. The final step in an erosion morphological operation, with a kernel size 3×3 . Erosion will remove small white noises in the boundary of the foreground [65].

In Figure 5-5 (a) represents the input and original image and (b) represents the resulting image after performing all the steps previously described.



Figure 5-5 Result of the mask generation of the retinal images.

5.2.1.2 IMAGE CROPPING

The cropping step is the step where most black borders and background is removed to potentiate shorter processing times and avoid useless information for CNN neural networks.

For this phase, in Appendix A, the original images and the masks were used, where the mask was used to find the contour of the circular retina (*cv2.findContours*) and form an area around this contour (*cv2.contourArea*).

Then a rectangular bounding box was added to the original images, going in search of the edges of the retina, obtained by the mask. As a starting point, the point (0,0) is in the upper left corner and is in the x —direction, in the first column, with a nonzero sum on lines corresponding to the left side of the foreground of the retinal image. Similarly, the process starts in the right direction of the image mask, in the y -direction, which allows finding the right side of the foreground of the retina image. This procedure is also replicated from the top and bottom of the image mask, obtaining the $y + h$ coordinates for the top and the $x + w$ coordinates for the lower part of the retina image, where h stands for height and w for width.

Figure 5-6 shows the results of the cropping process and the code excerpt corresponding to this function is present in Appendix A-1.

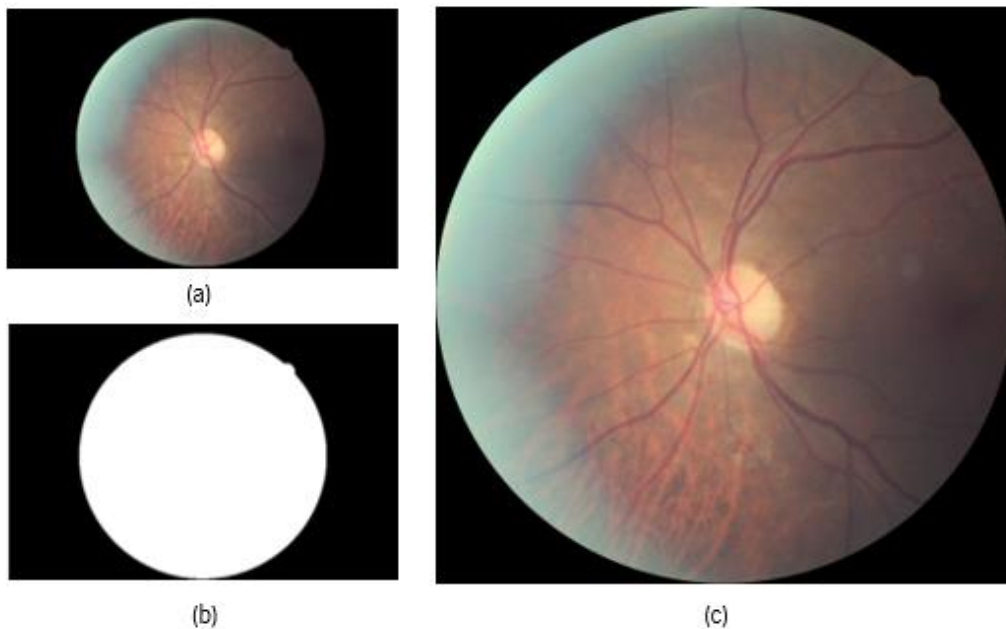


Figure 5-6 Result of the cropping process, (c) is obtained by adding a bounding box in the original image in (a), through an area obtained by the mask image in (b).

5.2.1.3 IMAGE RESIZING

The retinal images have different dimensions, after the cropping stage and have various forms of acquisition, in which the circle corresponding to the retina is totally in the image, and in other cases, the retina does not contain the upper and lower part of the retina as can be seen in Figure 5-7 (a). In Figure 5-6 (a), it is shown an example of a retina with the circular shape in its entirety. Due to these differences, it was thought to construct a function that to maintain the aspect ratio and make all the images have the same dimension, were added a black frame in the background, with the image centered, as it can be found in Figure 5-7 (c). The first step would be to define an aspect ratio given by dividing the width by height and introducing the variables x and y , which x is respectively the left-to-right direction and y corresponds to the top-to-down direction. Then *fill_color* will fill in the black color at the top and bottom of the image, like the image (a) of Figure 5-7. To all the images like (a) of Figure 5-6 there is not added a frame because it contains the complete circle and will not tend to lose their appearance, that is, they do not suffer from shrinking or stretching. All retinal images have been resized to 512 x 512 pixels.

The code excerpt corresponding to this function is present in Appendix A.1.

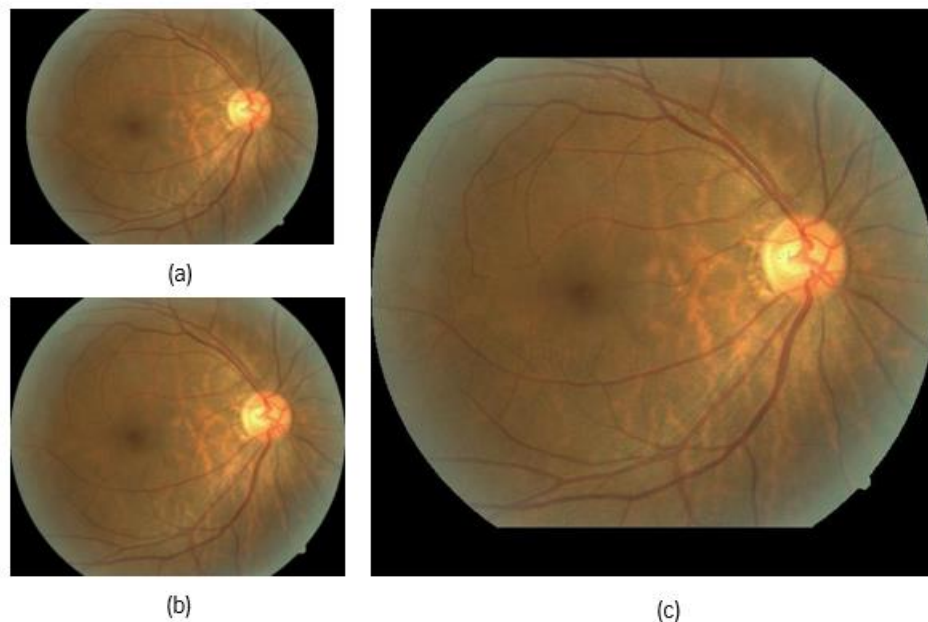


Figure 5-7 Result of the resizing process. The dimensions of the original image in (a) is 2560x1920 pixels; in the cropped image (b) 2306x1920 pixels and in (c) 512x512 pixels.

5.2.2 DATA PREPARATION

5.2.2.1 TRAIN, VALIDATION AND TEST SPLIT AND LOADING DATA TO CSV FILES

In order for retinal images to be entered into CNN networks, they are divided into 3 subsets: a training set, validation set, and testing set:

- Train Set is the subset of data used to train and fit the model. The model “sees” and learns from this dataset.
- Validation Set is the set of images used to provide an unbiased evaluation of a model fit on training dataset while tuning model hyperparameters. In Deep Learning it's often included to prevent overfitting during the training process [33].
- Test Set is the set of data used to provide an unbiased evaluation of the model and is used when a model is completely trained with the train and validation sets.

Figure 5-9 shows how the subsets were split was made, in this work, with the use of a library called Scikit-learn [66] that is an open source library, widely used in Machine Learning and is implemented in Python, is possible to split arrays or matrices, with the *train_test_split* function. This function reads a file of type CSV (Comma-Separated Values) that contains the path of the images and respective classes to which they belong, and according to a test splitting rate, it divides into two subsets of images. The excerpt of the code is in Appendix A.2.

The first splitting, seen in Figure 5-8, was done between the train and test subsets, 20% of test images and 80% of train images, of which 80%, 20% would be validation and 80% of the train. With the use of an integer random seed, and this being equal for the two subsets divisions, it guarantees that when the dataset is split, the subsets are always the same allowing reproducibility between experiments.

Appendix B.1 graphs are presented with the distribution of focus images, by training, validation and test sets.

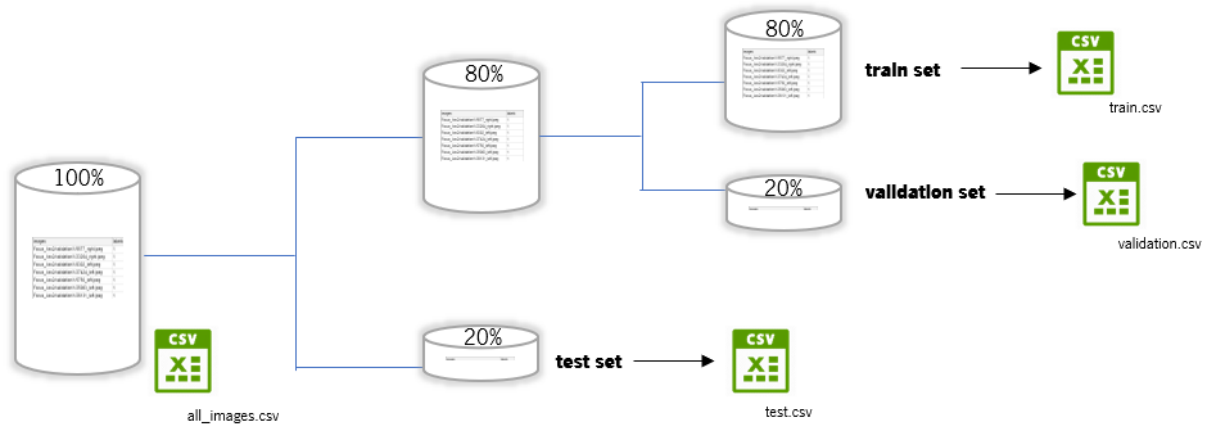


Figure 5-8 Diagram that explains the proportions of the train, validation and test sets split used in every CNN train and test.

After splitting the images into training, testing and validation, the two columns - directories of the images and their respective manual quality classification are read and placed again in three different CSV files, designated by train.csv, validation.csv and test.csv as shown in Figure 5-8.

For the contents of CSVs to be viewed and read on a JupyterLab notebook, the Pandas library was used. The data were kept in a DataFrame for their visualization, but for loading them into the CNN network, they were loaded and manipulated as numpy arrays.

As can be seen in Figure 5-9, the images were loaded from a certain CSV file, obtained in the previous stage (train.csv, validation.csv and test.csv). This CSV contains the path of the location of the image and its label or classification given by the human.

```
import pandas as pd

df = pd.read_csv('Focus_Ass2/train.csv', header=None)
pd.DataFrame(df.values[:, :])
```

| | images | labels |
|--|-------------------------------------|--------|
| | Focus_Ass2/train/1/22719_right.jpeg | 1 |
| | Focus_Ass2/train/1/8728_right.jpeg | 1 |
| | Focus_Ass2/train/1/18105_left.jpeg | 1 |
| | Focus_Ass2/train/1/7370_left.jpeg | 1 |
| | Focus_Ass2/train/1/27759_right.jpeg | 1 |
| | Focus_Ass2/train/1/15884_left.jpeg | 1 |
| | Focus_Ass2/train/1/9610_right.jpeg | 1 |
| | Focus_Ass2/train/1/27160_left.jpeg | 1 |
| | Focus_Ass2/train/1/7224_left.jpeg | 1 |
| | Focus_Ass2/train/1/32396_right.jpeg | 1 |
| | Focus_Ass2/train/1/44281_left.jpeg | 1 |

Figure 5-9 Reading of a CSV file with Pandas DataFrame, in a Jupyter Notebook.

5.2.2.2 DATA NORMALIZATION

Before the images were loaded onto the CNN network, they were normalized.

The normalization used linear normalization, where the image intensities are in a range of intensities of [0-1], obtained by dividing the respective pixel level of the image by the highest intensity value that an image can have (255), in a range of values of [0-255] (0 - minimum and 255 - maximum). The standardization function used can be found in Appendix A.2.

Normalization has been reported by the literature to improve performance. Sola and Sevilla [67] pointed out the importance of data normalization prior to the neural network training to fasten the calculations and obtain good results.

It wasn't applied any other preprocessing that would alter the image quality since the aim of this work is to assess the quality of the original images, in RGB e and with pixel intensity between 0 and 1.

5.2.3 CONVOLUTIONAL NEURAL NETWORKS

5.2.3.1 CNN DESIGN AND TOPOLOGY

For the present work, two CNN architectures were created and studied, one based on the AlexNet architecture present in the article [25] and also inspired by the shallowNet in [25] and the other network was not inspired in the state-of-the-art architectures and papers. The first developed network is called Net1 and the second is Net2.

To create and design these CNN networks, some guidelines and rules were followed. First, all the hidden layers (fully connected layers) should have the same number of neurons per layer. Second, typically, two hidden layers are good enough to use in a neural network, once, it solves the majority of problems. Using scaling or batch normalization, with mean=0 and variance=1, for all input variables after each layer could improve the convergence effectiveness, and finally, step size reduction after each epoch/iteration could improve convergence, in addition to the use of momentum and dropout [36].

AlexNet, presented in Figure 5-10, is an architecture that consists of 5 convolutional layers, 3 pooling layers, 2 local response normalization layers and 2 fully connected layers. The shallowNet has three convolutional layers with 96, 256 and 256 convolutional filters (Figure 5-11)

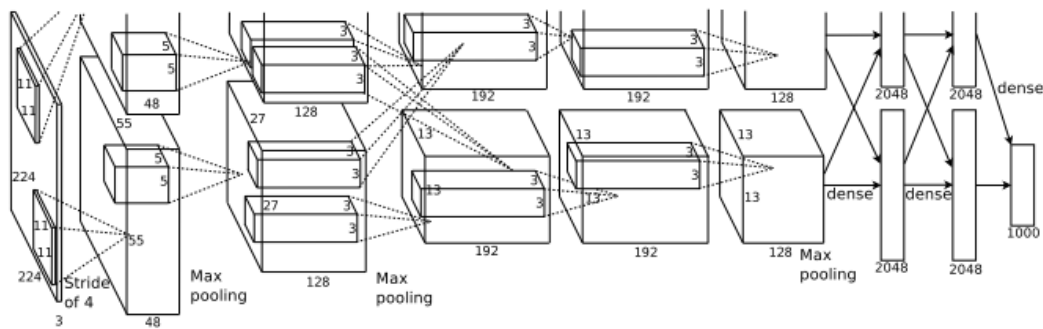


Figure 5-10 AlexNet architecture (adapted from [25]).

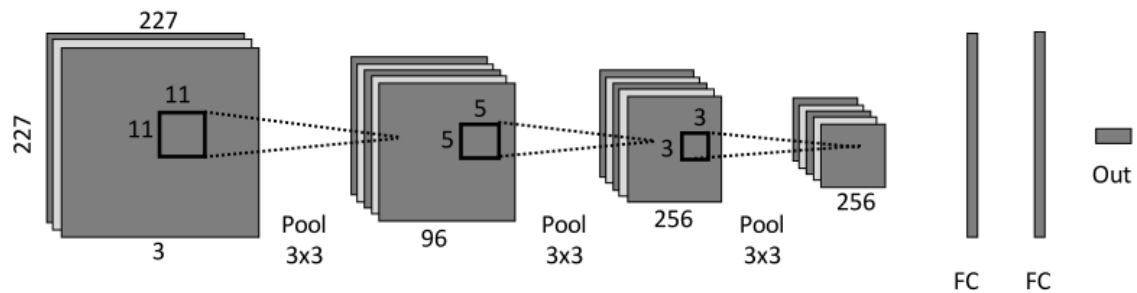


Figure 5-11 shallowNet architecture (adapted from [24]).

Net1 has five convolutional layers, with 4, 9, 12, 12 and 9 convolutional filters of size 11x11, 5x5, 3x3, 3x3, 3x3 and 2 fully connected layers, with 200, 100 neurons respectively and an output/classification layer. In case of Binary Classification, it was used a Sigmoid Unit in conjunction with binary cross entropy loss function. In case of multiclass classification, it was used a softmax unit in conjunction with categorical cross entropy loss function.

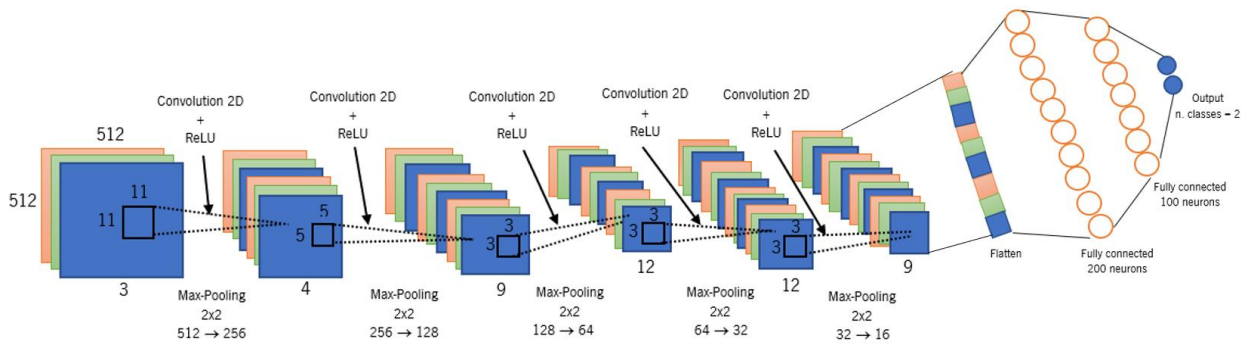


Figure 5-12 Net1 neural network.

Net2 has also five convolutional layers, with 11, 11, 22, 22 and 44 convolutional filters of size 3x3. A configuração da rede inspirada na AlexNet, Net1, é apresentada de seguida, na Figura 5-12 e a rede Net2 na Figura 5.13.

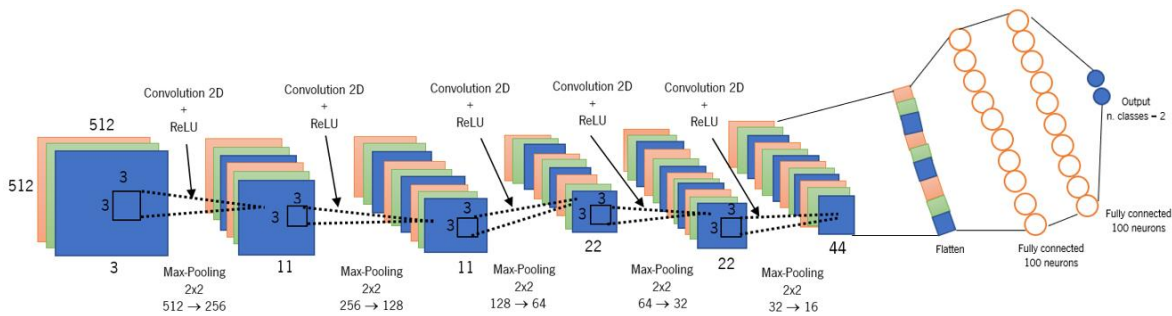


Figure 5-13 Net2 neural network.

In Net1 and Net2 networks, the activation functions used after each convolution were the Rectified Linear Units (ReLU), which introduce nonlinearity to the system; without these the model would only learn linear mappings that are present in the convolution operations [68].

The choice of this activation function followed the recent studies conducted by researchers who have found that these activation layers have the ability to accelerate the convergence of the SGD optimization function in relation to tanh and sigmoid activation functions, without creating change in accuracy [25] and without modifying the scale of the input images [26].

Both in Net1 and Net2 networks, the input images took as spatial resolution, 512x512 pixel dimensions that follow a 2-Dimensional convolution layer. The filter/kernel size of the first convolutional layer was 4x4x3, in the case of Net1 (width = 4, height = 4 and 3 RGB channels) and a filter size of 11x11x3 in the case of Net2 (width = 11, height = 11, RGB channels = 3). The pooling layers, where the layers that followed each of the convolution layers, downsampling the spatial dimensions of the outputs of each convolution, independently in each depth slice of the input volume. Both Net1 and Net2 followed the CONV-POOL configuration, five blocks throughout the network.

It was pre-defined in the Keras data format options, that the input size would be of the *channels_last* type, for example, 512x512x3 with width = 512, height = 512 and channels = 3, as can be seen in Appendix A.3.

The input images had the dimensions 512x512x3, (width=512, height=512, channels=3, 3 channels because the input volume is RGB), and the pooling implemented throughout the network was max-pooling with a 2x2 filter and a stride=2 (slides 2 in 2 pixels of the input image).

This max-pooling layer is responsible for the downsampling and reduction of spatial resolution of input image by half; this means, in each pooling layer, the output dimensions are half of the input dimensions from the last convolution layer.

Following each convolution layer, the image is halved five times, having an original dimension of 512x512 pixels and ending with 16x16 pixels. These 16x16 pixel feature maps are then fed and converted to a 1-Dimension vector to be understandable for the following layers (the fully connected layers), combining all the found local features of the previous convolutional layers into one layer.

After choosing the kernel and stride size, the padding parameter chosen was zero-padding or "same", to prevent that the image width is not shrunk to a pixel less than the kernel width at each layer and, therefore, to prevent the loss of edge information. This type of padding allows greater control over the kernel width and the size of the output independently.

Table 5-2 summarizes the values agreed for each parameter in the Net1 and Net2 networks.

Table 5-2 Summary of all the parameter values of CCN topology.

| | Net1 | Net2 |
|-----------------------------|------------------------------|--------------------|
| Number of kernels | 4, 9, 12, 12, 9 | 11, 11, 22, 22, 44 |
| Kernel size | 11x11, 5x5, 3x3, 3x3 and 3x3 | all 3x3 |
| Padding | Zero-padding | Zero-padding |
| Stride | 2x2 | 2x2 |
| Max-pooling | 2x2 | 2x2 |
| Number of FC neurons | 200, 100, (1 or 3) | 100, 100, (1 or 3) |

After defining all parameters discussed above, in the fully connected layers weights and bias initializers were added, as you can see an excerpt from the script with those of the Net2 FC layers, in Figure 5-14.

```

model.add(Dense(100,
                activation = "relu",
                kernel_constraint = maxnorm(3),
                kernel_initializer = 'uniform',
                bias_initializer = 'zeros'))
model.add(Dropout(0.5))
model.add(Dense(100,
                activation="relu",
                kernel_constraint = maxnorm(3),
                kernel_initializer = 'uniform',
                bias_initializer = 'zeros'))
model.add(Dense(1,
                activation='sigmoid',
                kernel_initializer = 'uniform',
                bias_initializer = 'zeros'))

```

Figure 5-14 Weights and bias initializations.

Network weights are initialized with uniform distributed random real values ranging between 0 and 1, and the experiments done in Net1 and Net2 use the same initial random weights method.

According to Shin et al. [42], arbitrary weights initialization can delay or stop convergence and delay the back propagation process, so a correct initialization of weights can mean convergence for neural networks. The typical configuration of the structure of a sequential CNN network is shown in figure 5-15, which shows the layer name, the output size of each layer and the number of parameters of each layer. In this particular case, figure 5-15 represents the network topology Net2, with the 5 convolution-maxpooling blocks, the flatten layer and the 3 fully connected layers. The total number of parameters of the network, is just below the last FC layer description.

This model can be obtained by the *model_print* command in a Jupyter Notebook cell. The models topology Net1 and Net2 are in Appendices C.1 and C.2.

RETINAL QUALITY ASSESSMENT EXPERIMENTS

| Layer (type) | Output Shape | Param # |
|-------------------------------|----------------------|---------|
| conv2d_11 (Conv2D) | (None, 512, 512, 4) | 1456 |
| max_pooling2d_11 (MaxPooling) | (None, 256, 256, 4) | 0 |
| conv2d_12 (Conv2D) | (None, 256, 256, 9) | 909 |
| max_pooling2d_12 (MaxPooling) | (None, 128, 128, 9) | 0 |
| conv2d_13 (Conv2D) | (None, 128, 128, 12) | 984 |
| max_pooling2d_13 (MaxPooling) | (None, 64, 64, 12) | 0 |
| conv2d_14 (Conv2D) | (None, 64, 64, 12) | 1308 |
| max_pooling2d_14 (MaxPooling) | (None, 32, 32, 12) | 0 |
| conv2d_15 (Conv2D) | (None, 32, 32, 9) | 981 |
| max_pooling2d_15 (MaxPooling) | (None, 16, 16, 9) | 0 |
| flatten_3 (Flatten) | (None, 2304) | 0 |
| dense_7 (Dense) | (None, 200) | 461000 |
| dropout_3 (Dropout) | (None, 200) | 0 |
| dense_8 (Dense) | (None, 100) | 20100 |
| dense_9 (Dense) | (None, 1) | 101 |
| Total params: 486,839 | | |
| Trainable params: 486,839 | | |
| Non-trainable params: 0 | | |

Figure 5-15 Sequential model configuration of Net1.

5.2.3.2 CNN TRAINING – FITTING AND COMPILING MODEL

With all the layers added to the network, it is important at a later stage to compile, train and continually improve network performance. To compile the model, the `model_compile` command requires passing parameters such as the type of optimization function to use, the learning rate, and the loss function. After compiling it is necessary to fit the model by the `model_fit` command, and in this phase the batch size parameter is added, the number of training epochs, the loaded training and validation images and even add or not callbacks (such as Early Stopping and the Model Checkpoint that saves the best model). In Appendices A.3 is the code excerpt corresponding to these steps.

The optimization functions used in the experiments were:

- Adam, with LR ranging from 0.1 to 0.00001;
- SGD, with LR varying between 0.1 and 0.00001;
- SGD with momentum = 0.9 and varying between 0.1 and 0.00001.

The functions of loss used were:

- binary cross entropy, in the case of a binary classification with output equal to [0,1] or [1,0];
- categorical cross entropy, if it is a multi-class classification with categorical output equal to [0,0,1], [0,1,0] or [1,0,0].

The activation functions used were:

- ReLU in all convolutional layers;
- Sigmoid in the last fully connected layer when it came to binary classification;
- Softmax in the last fully connected layer when it was a multi-class classification.

The batch size varied between 4,12,24,32 and 64.

The number of epochs used was 400 epochs, since the use of 50, 100 and 200 epochs, meant that some training did not converge.

5.2.3.3 MODEL OPTIMIZATION AND HYPERPARAMETERS TUNING

Some effects like overfitting or the internal covariate shift may be present when a model is being created and trained. The techniques used to reduce or eliminates these effects are known as regularization and optimization methods.

To reduce and monitor overfitting the following techniques were tried:

- Add **validation set** to measure how much a model is generalizing, since these are images that the model will never see, and therefore will never memorize, the introduction of this set of images is a strong indicator that the network is not learning well and memorizes the training data (i.e. is overfitting) or if the model is learning the features of the training cases without memorizing (model is generalizing - good fitting);
- **Learning process "babysitting"** which is a way of seeing the progress of the results either at the graphic level (by the progress of the learning curves) or by making a print of the results using various techniques in order to see which parameters are important and makes the network learning better (grid search of techniques);
- **Grid search** that is directly associated to the validation set, since, when choosing the best combination of hyperparameters, both the validation training and the validation loss will have satisfactory values throughout the training. This method is obtained by choosing finite values that the hyperparameters (batch size, learning rate, ...) can assume [33].
- **Dropout** with probability = 0.5, between the second and third fully connected layer;
- **L_2 regularization** or weight decay with $\lambda = 0.001$. Were performed empirical tests for λ equal to 0.1, 0.01 and 0.001, obtaining the best global value of $\lambda=0.001$;
- **Early Stopping** with mode = 'auto', patience = 50 epochs and monitor = 'val_loss';
- **Max-norm** constraint with c=3;

In order to reduce the internal covariate shift phenomenon, a Batch Normalization layers were introduced between the activation layer and the max-pooling layer in each CONV-POOL block.

5.2.3.4 EVALUATION OF CNN PERFORMANCE

To evaluate the Net1 and Net2 models trained, the test set (unseen data) created before was used, along with the respective image labels for each image. Like train and validation sets, the test set had all the preprocessing steps (mask creation, crop and resize images with 512x512 pixels). Then, the linear normalization was performed.

After obtaining and saving the best model and weights by Model Checkpoint and CSV logger callbacks, this model was loaded with the test dataset with *model_predict*. Computing this last command, the classifications were performed, with the predicted labels returning an array with the probability of each class and a function called *argmax*, giving the one-hot vector classifications [0,1], [1,0] (for binary classification) and [0,0,1], [0,1,0], [1,0,0] (for multi-class classification).

With the Python module, *model_print_predictions*, was possible to obtain a list of the TP, FP, TN and FN, and to know exactly which the incorrect classifications were.

These results were given by a log file, that stored all the TP, FP, TN and FN values, and automatically, calculated the sensitivity, specificity, accuracy, precision, F_1 -score and AUC, having the advantage to visualize the log files anytime.

Both predicted and ground truth labels are compared, and a confusion matrix is created and plotted. Also, ROC curve, AUC and the classification computation time were given. The code used to test and evaluate the model performance and give the list of the classification predictions are in Appendices A.4.

```
02/10/2018 11:10:13 - INFO: TP:145 TN:150 FP:10 FN:15
02/10/2018 11:10:13 - INFO: Accuracy:0.921875
02/10/2018 11:10:13 - INFO: Sensitivity:0.906250
02/10/2018 11:10:13 - INFO: Specificity:0.937500
02/10/2018 11:10:13 - INFO: Positive Predictive Value:0.937500
02/10/2018 11:10:13 - INFO: Negative Predictive Value:0.906250
```

Figure 5-16 Log file generated with the classification metrics.

6 RESULTS AND DISCUSSION

In this Chapter the classification results for the focus, color and illumination feature extraction are presented. The cases studied in the present work are binary classification for focus and illumination analysis and multi-class classification for color analysis. Focus parameter is classified as blurred or focused; illumination is classified as even or uneven; and finally, the color is classified in dark, normal or bright.

6.1 CLASSIFICATION MEASURES

In supervised Machine Learning, there are several ways to evaluate the performance of learning algorithms and the classifiers they produce. The measures for the classification model quality are built from a confusion matrix which records correctly and incorrectly recognized examples from each class [69].

The positive class is the occurrence of poor image acquisition (blurred, uneven illumination, dark and bright color) and the negative class is the occurrence of good image acquisition (even, focused and normal color).

The confusion matrix records the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) in binary classification as can be seen in table 6-1.

- True Positive (TP): correctly positive predicted classes, that are blurred images correctly predicted as blurred;
- False Positive (FP): incorrectly positive predicted classes, that are focused images predicted as blurred;
- True Negative (TN): correctly negative predicted classes, that are focused images correctly predicted as focused;
- False Negative (FN): incorrectly negative predicted classes, that are blurred images predicted as focused.

Table 6-1. Confusion matrix for binary classification.

| | | Predicted Class | |
|---------------------|----------|------------------------|----------|
| | | Negative | Positive |
| Actual Class | Negative | TN | FP |
| | Positive | FN | TP |

Multi-class classification assumes that each image is assigned to one and only one label. In the color parameter assessment context, an image can only be predicted as normal, bright or dark. For this kind of classification, as shown in table 6-2, it's also used a confusion matrix, where the elements in the main diagonal show the correct predicted classifications and all other elements, off the diagonal, are classified as incorrect.

Table 6-2. Confusion matrix for multi-class classification.

| | | Predicted Class | | |
|---------------------|---------|------------------------|---------|---------|
| | | Class 1 | Class 2 | Class n |
| Actual Class | Class 1 | TP | | |
| | Class 2 | | TP | |
| | Class n | | | TP |

With this confusion matrix, the overall evaluation metrics are averages across classes. The number of instances of each class in train, validation and test are important to consider, since the class unbalance can play a determinant role, of which kind of average to choose. There are two different averages:

- "macro-average" that calculates the mean of the binary metrics, giving equal weight to each class.
- "micro-average" this variant extends the classification measures to the averaged values across all the classes, by treating one class as negative and others as positives. However, this variant doesn't place emphasis on rare classes, and is not recommended to use, when the classes are imbalanced [70].

Accuracy is the most general way of comparing algorithms and the most used empirical measure used in binary and multiclass classification, without focusing on a class, but doesn't distinguish between the number of correct labels of different classes. The accuracy can be obtained through the values obtained by the confusion matrix and can be computed as follows, in equation 6.1.

$$accuracy_{binary\ class.} = \frac{TP + TN}{TP + FP + FN + TN} \quad (6.1)$$

In multi-class classification the total accuracy can be computed as follows in equation 6.2:

$$accuracy_{multi-class.} = \frac{TP_1 + TP_2 + TP_n}{\sum all\ cells\ in\ the\ CM} \quad (6.2)$$

Beyond accuracy, there are other metrics that evaluate the model's performance and their way of correctly predicting the positive class and the negative class. These metrics are sensitivity (SN), also called recall or true positive rate (TPR) (equation 6.3), specificity (SP) (equation 6.4), and precision (P) (equation 6.5) which confirm or refute the presence or absence of bad image acquisition.

Sensitivity and specificity are often employed in biomedical applications and in studies involved image and visual data [69].

$$sensitivity = \frac{TP}{TP + FN} \quad (6.3)$$

$$specificity = \frac{TN}{TN + FP} \quad (6.4)$$

$$precision = \frac{TP}{TP + FP} \quad (6.5)$$

Sensitivity or Recall (equation 6.3) is a metric often used in disease detection where it gives high classification scores, achieving high numbers of true positives and avoiding false negatives, that is, that rarely failed the occurrence of disease. To obtain a high sensitivity value it's necessary that

the classifier has predicted a high number of TP and a low value of FN (FN occurs when the patient definitely has a poor image acquisition, but the classifier evaluated as good quality).

Specificity (equation 6.4) measures the proportion of patients with good image acquisition that are correctly classified as not having bad acquisition, in other words, is the classification percentage of correctly rejecting the good acquisition images that are actually good.

Precision (equation 6.5), is another ML metric, where it's important to avoid false positives. This metric also shows how confident and accurate the classifiers are. To increase precision, the number of true positives, that the classifier predicts must be high or the number of false negatives must be low.

The combination of the recall and precision metrics result in a metric called F_1 score found in the equation 6.6.

$$F_1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6.6)$$

$F_1 \text{ score}$ is used as a classifier evaluation metric, when only the accuracy does not give the necessary information of the classifier performance.

Therefore, in the present work, it is intended that the classifier would give a high recall value, with a reduced number of false negatives, but also to predict when an event is positive when in fact it is, that is, it also has a high precision value.

Ideally, these metrics would show all patients with good image acquisition, and similarly, all patients who do not have good image acquisition.

A perfect test is never positive in a patient who has a good image acquisition and is never negative in a patient who has in fact bad image acquisition [71], because the evaluation of the classifier may contain some margin of error, this classifier should try to reduce the false negative predictions in that the patient definitely has a poor image acquisition, but the classifier evaluated as good quality.

Another of the problems that arise in the classification is class imbalance, when the datasets studied exhibits an unequal distribution between its classes [72]. For several base classifiers, studies have shown that, a balanced data set provides improved overall classification performance compared to an imbalanced dataset [73], [74].

For the present study it was not necessary to carry out class balancing techniques, since in the Data Acquisition process (section 5.1), the number of images corresponding to each class was considered, and a manual classification of images was carried out in such a way that there were a similar number of images classified for each class.

One way to graphically visualize the behavior of the false positive rate (FPR or specificity) and the true positive rate (TPR or 1-sensitivity) is called receiver operating characteristic (ROC), represented by Figure 6.1. This is presented as a curve, in the unit square, where the TPR value lies on the vertical axis and the FPR value lies on the horizontal axis. In the top left corner is the ideal point of the ROC curve, with the FPR at zero and the TPR at 1, as shown in Figure 6-1 in curve A.

The area under the curve (AUC) is the area underneath the ROC curve and is an effective and combined measure of TPR and FPR that describes the inherent validity and performance of the classifier [75]. When the AUC has a value of 0.5 it's called the random classifier and as the classifier improves its performance, the AUC also walks to higher values and closer to the optimal value.

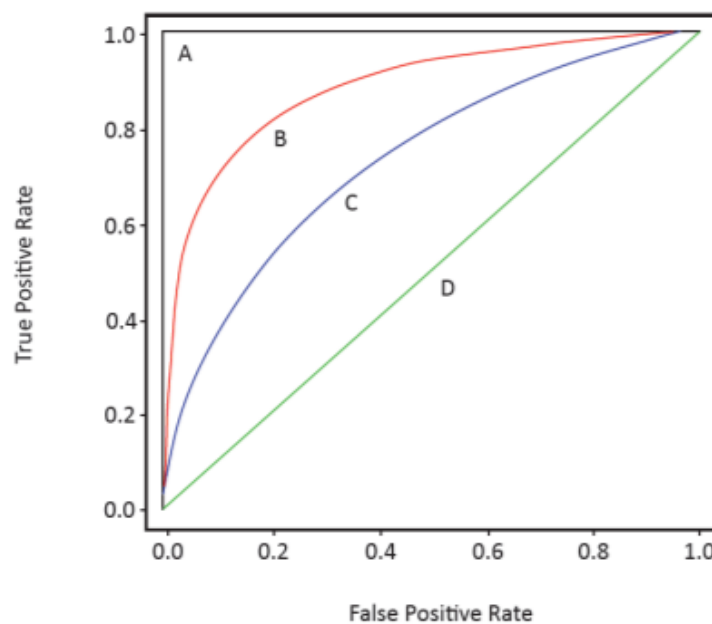


Figure 6-1 ROC Curves. Curve A represents and AUC=1 and a perfect test; curve B are a good and a moderate diagnostic results, respectively; and D is the random classifier, with an AUC=0.5. Adapted from [76].

6.2 CLASSIFICATION RESULTS OF FOCUS ASSESSMENT

The results of the focus parameter are divided between the results obtained from Net1 and Net2. A total of 378 images were used to test the two models, divided into 202 positive (unfocused) images and 176 negative (focused) images. For training and validation, 1510 and 472 images were used respectively. The distribution of the images by training, validation and test dataset can be found in Appendix B-1.

The loss function used was binary cross entropy, widely used in binary classification, and three different optimization functions were used: Stochastic gradient descent (SGD), Stochastic gradient descent with momentum (SGD + Momentum) and Adam.

The following regularization was applied to Net1 and Net2: dropout between the first and the second fully connected layers with probability of 0.5 and the weight decay regularization L^2 with factor of $\lambda = 0.001$.

The fixed and varied parameters used in the training of the two models are presented in table 6.3 and table 6.4.

Table 6-3 Fixed parameters used in each model Net1 and Net2.

| Fixed parameter | Value |
|----------------------|---|
| Batch size | 4 |
| Epochs | 400 epochs (Early Stopping, patience=50 epochs) |
| Convolution Layers | 5 |
| Loss Function | binary cross entropy |
| Activation function | ReLU (Convolution layers) and Sigmoid (FCL) |
| Dropout | 0.5 |
| L^2 regularization | 0.001 |
| Momentum | 0.9 |

Table 6-4 Varied parameters used in each model Net1 and Net2.

| Varied parameter | Value |
|------------------------|-----------------------------------|
| Learning rate (LR) | 0.1, 0.01, 0.001, 0.0001, 0.00001 |
| Optimization Functions | SGD, SGD + Momentum, Adam |

6.2.1 RESULTS OF MODEL NET1

The cases where the accuracy test was no greater than 46.56% and AUC was equal to 0.5 (random guessing), Batch Normalization was then implemented, in order to increase accuracy and other classification metrics. The results are shown in Table 6.5, 6.6 and 6.7 and with a star are marked the models that went well without Batch Normalization.

The abbreviations presented in the tables are: TP for True Positives, TN for True negatives, FP for False Positives, FN for False Negatives, SN for Sensitivity, SP for Specificity, P for Precision and AUC for Area under the curve. The values of sensitivity, specificity, precision and F_1 -score are in percentages and the AUC value is between 0 and 1.

Table 6-5 Classification performance values for each model trained with each LR and optimization function, without Batch Normalization.

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|------------|--------------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.1 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.01 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.001 | 96.56 | 192 | 173 | 3 | 10 | 95.05 | 98.30 | 98.46 | 0.97 | 96.73 |
| | 0.0001 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.00001 | 92.06 | 186 | 162 | 14 | 16 | 92.08 | 92.04 | 93.00 | 0.92 | 92.54 |
| SGD | 0.1 | 82.54 | 155 | 165 | 11 | 47 | 76.73 | 93.75 | 93.38 | 0.85 | 84.24 |
| | 0.01 | 96.83 | 193 | 173 | 3 | 9 | 95.54 | 98.30 | 98.47 | 0.97 | 96.68 |
| | 0.001 | 97.62 | 197 | 172 | 4 | 5 | 97.52 | 97.73 | 98.01 | 0.98 | 97.77 |
| | 0.0001 | 95.77 | 190 | 172 | 4 | 12 | 94.06 | 97.73 | 97.94 | 0.96 | 95.96 |
| | 0.00001 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| SGD + Mom. | 0.1 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.01 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.001 | 96.30 | 192 | 172 | 4 | 10 | 95.05 | 97.73 | 97.96 | 0.96 | 96.48 |
| | 0.0001 | 93.03 | 189 | 174 | 2 | 13 | 93.56 | 98.86 | 98.85 | 0.96 | 96.18 |
| | 0.00001 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |

The results of table 6.7, with Batch normalization, show significant improvements comparing with table 6.5, in which the classifier was able to learn very well the training, validation and test cases, as can be seen in figure 6.2 where it shows training with batch normalization. In some cases, with the same learning rate and optimization function, without batch normalization, the classifier did

not obtain better accuracy than the 46.56%, and an AUC greater than 0.50, and therefore could not learn or extract features from the images.

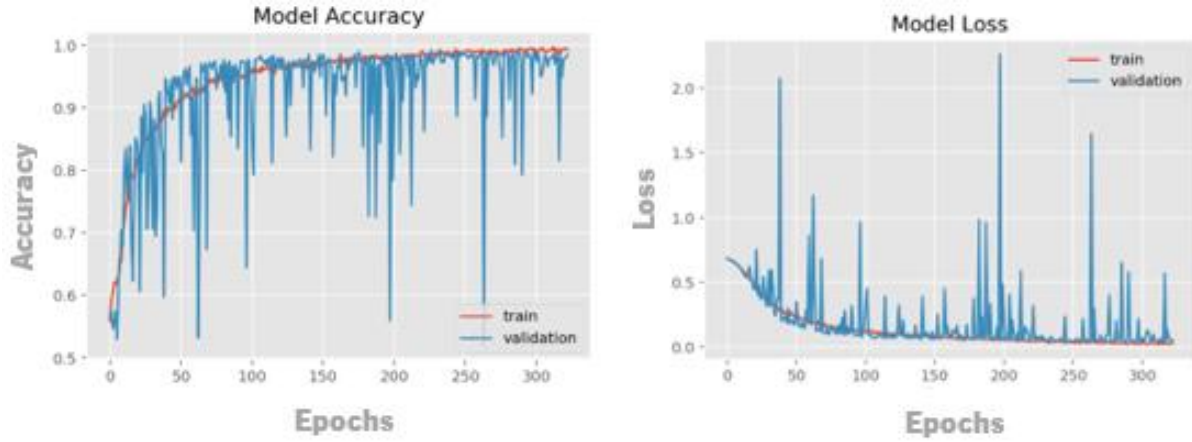


Figure 6-2 Accuracy and Loss learning curves of the trained network, with Batch normalization, for LR=0.0001 and the ADAM optimization function.

To reduce the validation learning curve instability, of accuracy and loss, shown in figure 6.2, the dropout layer was removed. Figure 6.2 corresponds to training with dropout and figure 6.3, the training without dropout.

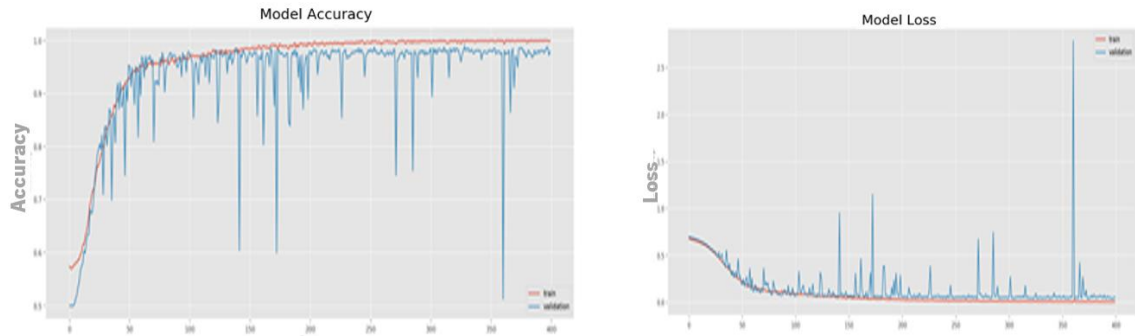


Figure 6-3 Accuracy and Loss learning curves of the trained network, with Batch normalization and without the dropout layer, for LR=0.0001 and the ADAM optimization function.

By the table 6.6 visualization, it was found that although in training C was no improvement in the test accuracy value or in the value of F_1 -score, the model created in C detected much less FN compared to A and B, and so it is a better sensitivity classifier, and therefore, goes in the right way to the intended classifier.

RESULTS AND DISCUSSION

Table 6-6 Training performed for the LR = 0.0001 and the ADAM optimization function, where A represents the training without BN, B represents training with BN and C represents the training with BN and without dropout.

| | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|----------|-----------|-----|-----|----|-----|-------|-------|-------|------|--------------|
| A | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| B | 97.09 | 193 | 174 | 2 | 9 | 95.54 | 98.86 | 98.97 | 0.97 | 97.22 |
| C | 96.56 | 198 | 167 | 9 | 4 | 98.02 | 94.89 | 95.65 | 0.96 | 96.82 |

Although the removal of the dropout layer has reduced the instability of the accuracy and loss learning curves, there was a greater tendency of the validation curve to disperse from the training curve and cause a slight overfitting, as can be seen in figure 6.3. There were no significant improvements, in test accuracy and all other metrics, except for FN detection, which has a lower value than the classifier trained only with Batch normalization, and therefore greater sensitivity (sensitivity=98.02%). It was also noted that removing the dropout layer did not improve the results, so all the next models were trained with the dropout layer.

Table 6-7. Final frame with the all the best classification metric values for each LR and optimizer studied. With a star are marked the models that performed better without Batch Normalization.

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|------------|---------------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.1 | 46.56 | 0 | 176 | 0 | 202 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.01 | 95.77 | 192 | 170 | 6 | 10 | 95.05 | 96.59 | 96.97 | 0.96 | 96.00 |
| | 0.001* | 96.56 | 192 | 173 | 3 | 10 | 95.05 | 98.30 | 98.46 | 0.97 | 96.73 |
| | 0.0001 | 97.09 | 193 | 174 | 2 | 9 | 95.54 | 98.86 | 98.97 | 0.97 | 97.22 |
| | 0.00001* | 92.06 | 186 | 162 | 14 | 16 | 92.08 | 92.04 | 93.00 | 0.92 | 92.54 |
| SGD | 0.1* | 84.66 | 155 | 165 | 11 | 47 | 76.73 | 93.75 | 93.38 | 0.85 | 84.24 |
| | 0.01* | 96.83 | 193 | 173 | 3 | 9 | 95.54 | 98.30 | 98.47 | 0.97 | 96.68 |
| | 0.001* | 97.62 | 197 | 172 | 4 | 5 | 97.52 | 97.73 | 98.01 | 0.98 | 97.77 |
| | 0.0001* | 95.77 | 190 | 172 | 4 | 12 | 94.06 | 97.73 | 97.94 | 0.96 | 95.96 |
| | 0.00001 | 93.12 | 186 | 166 | 10 | 16 | 92.08 | 94.31 | 94.90 | 0.93 | 93.47 |
| SGD + Mom. | 0.1 | 57.94 | 82 | 137 | 39 | 120 | 40.59 | 77.84 | 67.77 | 0.59 | 50.77 |
| | 0.01 | 96.03 | 197 | 166 | 10 | 5 | 97.52 | 94.32 | 97.46 | 0.96 | 96.24 |
| | 0.001* | 96.30 | 192 | 172 | 4 | 10 | 95.05 | 97.73 | 97.96 | 0.96 | 96.48 |
| | 0.0001* | 96.03 | 189 | 174 | 2 | 13 | 93.56 | 98.86 | 98.85 | 0.96 | 96.18 |
| | 0.00001 | 96.03 | 194 | 169 | 7 | 8 | 96.04 | 96.02 | 96.52 | 0.96 | 96.28 |

After obtaining all the results, it can be seen from Table 6.7 that the optimization function with the best classification results was the SGD, with the test accuracy ranging from 82.54% to 97.62%.

The best result obtained in the whole training pipeline is highlighted in bold in table 6.7, for an LR=0.001 and the SGD optimization function - test accuracy was equal to 97.62%, an AUC of 0.98, false negative number of 5 and number of false positives of 4.

For the same parameters, the best model was obtained, with Batch Normalization. This training was performed to verify if using Batch Normalization could reduce the rate of FN and FP and it was verified the opposite, since these increased considerably, thus reducing the sensitivity of the classifier. All other performance metrics for the classifier also decreased, showing why the use of Batch Normalization did not add improvements to the best classifier that had been previously obtained without Batch Normalization.

Table 6-8 Results obtained from the best classifier trained without Batch Normalization (A) and with Batch Normalization (B).

| | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1-score |
|----------|------------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|------------|-------------------------------|
| A | 97.62 | 197 | 172 | 4 | 5 | 97.52 | 97.73 | 98.01 | 0.98 | 97.77 |
| B | 85.98 | 158 | 167 | 9 | 44 | 78.22 | 94.89 | 94.61 | 0.87 | 85.64 |

The results in table 6.7 and table 6.8 (A) show that Net1 has achieved high accuracy (9 errors - 4 FP and 5 FN) over 378 images, with 97.52% of sensitivity (the probability that Net1 would correctly identify a blurred image) and a specificity of 97.73% (the probability that Net1 would correctly identify a focused image). Table 6.9 presents the confusion matrix for the best focus classifier and Figure 6.4 has the ROC curve with AUC value. The ROC curve was generated with three points by the sklearn *plot_ROC* function.

Table 6-9. Confusion matrix for the best focused images classifier.

| | | Predicted Class | |
|---------------------|---------|------------------------|---------|
| | | Focused | Blurred |
| Actual Class | Focused | 172 | 4 |
| | Blurred | 5 | 197 |

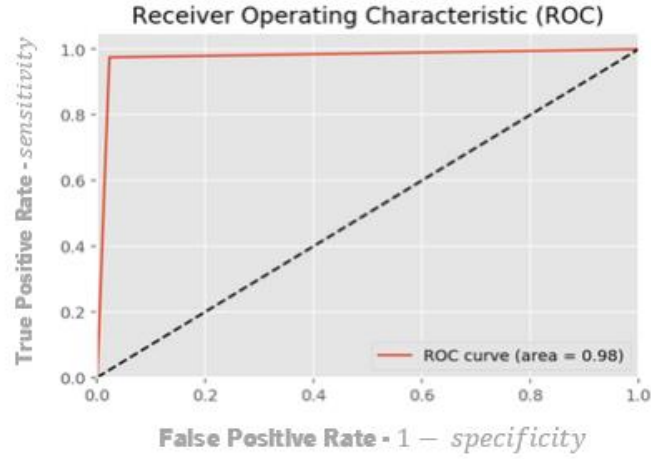


Figure 6-4 ROC curve for the best classification model, with LR=0.001 and SGD optimization function.

To obtain a qualitative idea of Net1's performance, the images that were erroneously classified are shown below (wrongly classified as focused and blurred). Figure 6.5 shows the images that are classified by the human as blurred class, but were classified as focused by the network, and Figure 6.6 shows images that are focused but classified as blurred by the network.

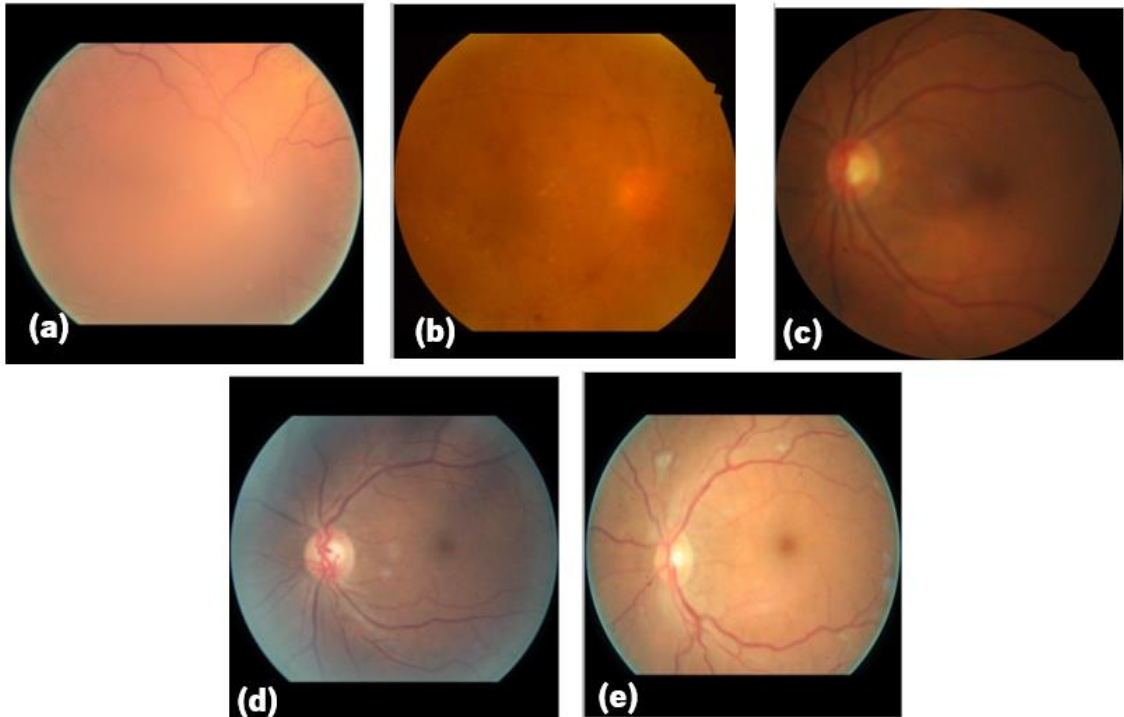


Figure 6-5 Images that represent instances of wrong blurred predicted images. a), b), c), d) and e) are labeled as blurred but predicted as focused. However, it can be concluded that d) and e) were correctly classified by the network as focused and incorrectly manually classified by the expert. This was a success case in which the network learned the features that differentiate between focus and blur.

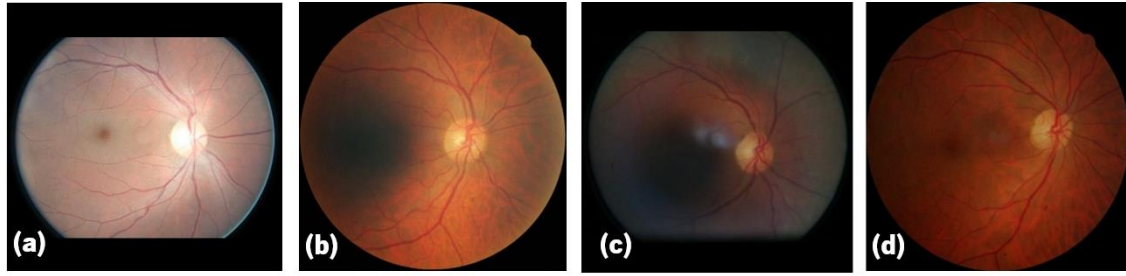


Figure 6-6 Images that represent instances of wrong focused predicted images. a), b), c) and d) are labeled as focused but predicted as blurred. It can be concluded that a) and d) are definitely focused, b) is a little blurred in the fovea area and c) is blurred and is a case of bad manual classification by the expert. Once again, the network learned the main differences between focused and blurred images.

After verifying and correcting the number of images correctly classified by the network Net1, in the test dataset, and that surpassed the human classification, all the metrics involved in the classification were recalculated, obtaining the values of table 6.10. Case A corresponds to the values before recalculation and those in case B are recalculated.

Table 6-10 Comparison of results before and after recalculation of all classification metrics.

| Case | Test Acc. | TP | TN | F | F | SN | SP | P | AUC | F_1 -score |
|----------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| A | 97.62 | 197 | 172 | 4 | 5 | 97.52 | 97.73 | 98.01 | 0.98 | 97.77 |
| B | 98.68 | 199 | 174 | 2 | 3 | 98.51 | 98.86 | 99.00 | 0.99 | 98.75 |

All values of the metrics in case B improved, obtaining a very low classification error, equal to 100% - 98.68% = 1.32% and a detection of FN = 3 of 5 images incorrectly classified as belonging to the class "focused "; and FP = 2 where previously 4 images were incorrectly classified as the " blurred "class.

RESULTS AND DISCUSSION

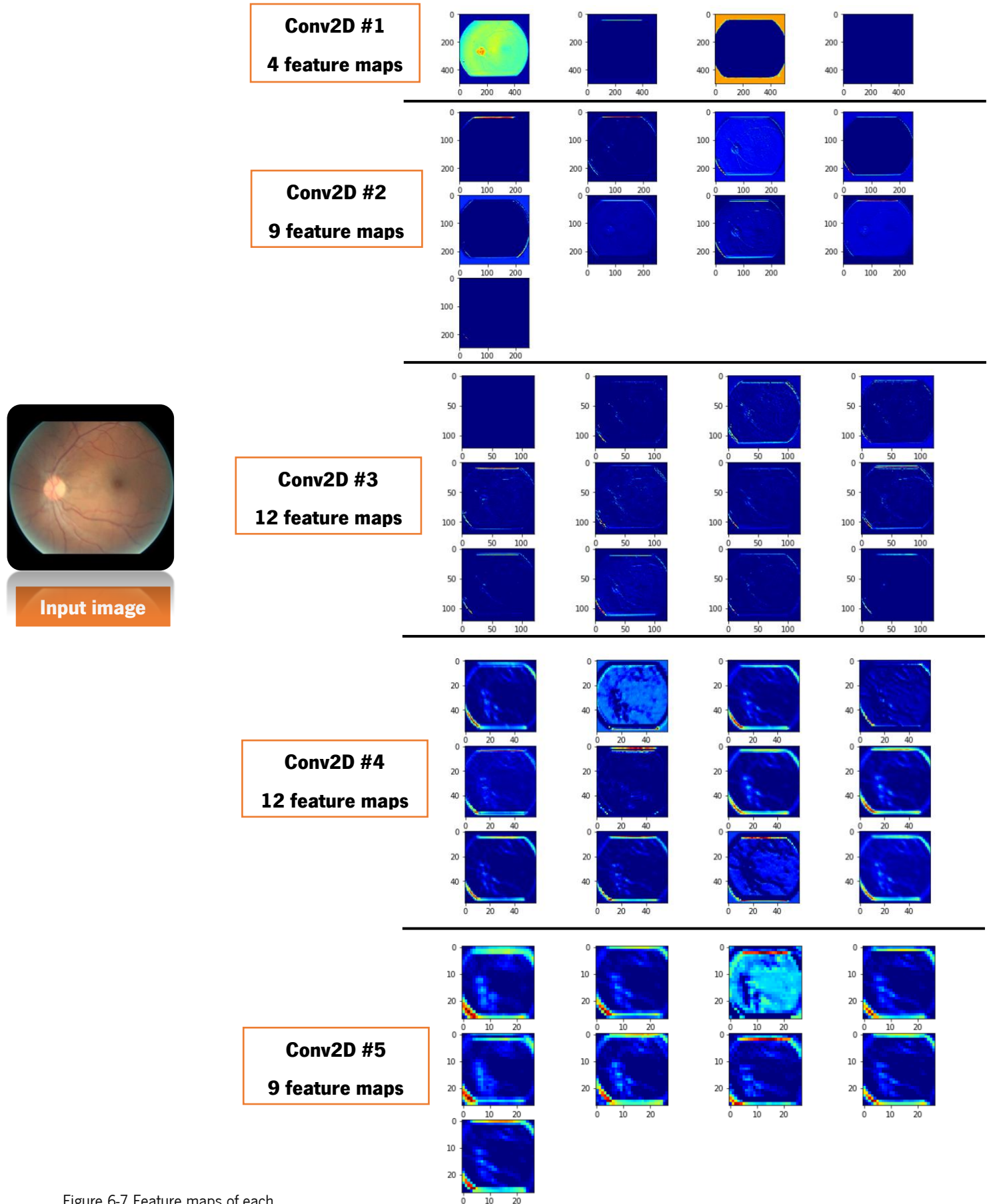


Figure 6-7 Feature maps of each convolutional layer for Net1.

CNN networks are seen as hierarchical feature extractors, where at the first level of extraction, from the first layers, simple features such as edges or local contrast are extracted. In the following layers of convolution more complex features are extracted that combine the several low level features [24]. The characteristics and information learned in the network through the feature maps for a blurred test image are represented in the images in figure 6.7.

In the Conv2D layer #1, 4 feature maps are generated that extracted geometric and border information between the retina and the background, as well as the differentiation of the retinal circle and the optic disc. Between the Conv2D #1 and Conv2D #2 layers there was a max-pooling operation that downsampled the image size in the first convolution layer to reduce by half the previous pixel value. In the second convolution layer (Conv2D #2), information about the location and volume of vessels and retinal constituents, such as the macula, optic disc and fovea, were mostly extracted. From Conv2D #3 to Conv2D #4 information is extracted from the color variation in the retina and the blur of the image with segmentation of blurred and focused areas. These segmentations are most prominent in Conv2D #5 with 9 feature maps of 32x32-pixel sized images.

The computation time for Net1 to train and extract was between 15min34sec and 1h50min4sec, depending on the optimization function and learning rate used. To predict the classes of 378 images, the execution time was between 1sec2msec and 3sec1msec.

The training that took the longest time was 1h50min and each training was about to be trained between 52 and 389 epochs/iterations. When Early Stopping was introduced, which contained a patience equal to 50 epochs, the training was extended while the reduction criterion of validation loss was maintained. The algorithm terminates when no parameters have improved over the best recorded validation error for some pre-specified number of iterations.

This regularization strategy proved to be a good parameter picker, returning models with the lowest validation loss value. After all the training sessions, it is verified that as the rate of learning rate decreases, the number of training periods increases, since the network requires more training epochs until it converges.

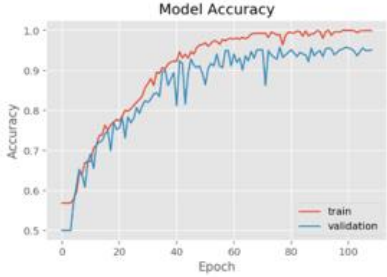

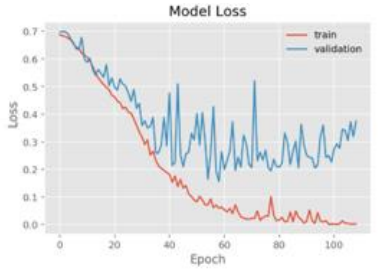

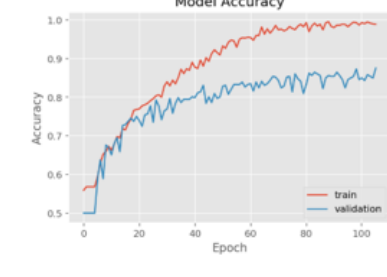
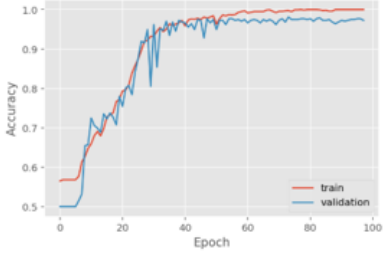
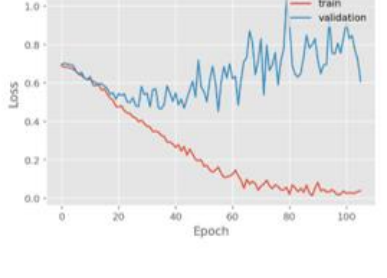
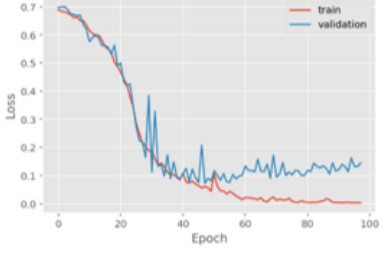
6.2.2 RESULTS OF MODEL NET2

Since there were good results in the use of Batch Normalization in the Net1 model, this method was also used in the Net2 training.

Table 6-12 has the Batch Normalization results without regularization L^2 (weight decay) and table 6-13 has the results of training with Batch Normalization with regularization L^2 . Was experimented the use of weights regularization L^2 , due to the appearance of the overfitting and instability phenomenon in some Batch Normalization accuracy and loss learning curves, in both training and validation states.

Table 6-11. the training curves are shown only with Batch Normalization and the training with Batch Normalization and with regularization L^2 .

Table 6-11 Learning curves of models trained with Batch Normalization and L2 regularizer, for two different LR and SGD optimization functions (trained with and without momentum).

| | Without Regularization L^2 | With Regularization L^2 |
|------------------------------|---|--|
| LR = 0.01 SGD |  |  |
| |  |  |
| LR=0.001 SGD + Mom. |  |  |
| |  |  |

By the validation loss learning curve visualization, Table 6-11 shows that there is a greater stabilization of learning by reducing of the peaks occurrence and the continuously reduction of validation loss values throughout the training, converging to smaller values and closer to zero.

The lines highlighted in the next tables (6-12 and 6-13) are related to the trained models with the best values and performance in the classification metrics.

RESULTS AND DISCUSSION

Table 6-12 Results obtained without the use of regularization L^2 .

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|------------|--------|--------------|-----------|-----------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.1 | 60.58 | 186 | 43 | 13 | 16 | 92.08 | 24.43 | 58.31 | 0.58 | 71.40 |
| | 0.01 | 96.83 | 19 | 17 | 6 | 6 | 97.03 | 96.03 | 97.03 | 0.97 | 97.03 |
| | 0.001 | 97.09 | 194 | 173 | 3 | 8 | 96.04 | 98.30 | 98.48 | 0.97 | 97.24 |
| | 0.0001 | 97.09 | 19 | 17 | 4 | 7 | 96.53 | 97.73 | 97.99 | 0.97 | 97.26 |
| | 0.0000 | 93.39 | 187 | 166 | 10 | 15 | 92.57 | 94.32 | 94.92 | 0.93 | 93.73 |
| SGD | 0.1 | 73.02 | 146 | 130 | 46 | 56 | 72.28 | 73.86 | 76.04 | 0.73 | 74.11 |
| | 0.01 | 94.44 | 188 | 169 | 7 | 14 | 93.07 | 96.03 | 96.41 | 0.95 | 94.71 |
| | 0.001 | 97.09 | 194 | 173 | 3 | 8 | 98.04 | 98.30 | 98.48 | 0.97 | 97.24 |
| | 0.0001 | 95.50 | 191 | 170 | 6 | 11 | 94.55 | 96.59 | 96.95 | 0.95 | 95.74 |
| | 0.0000 | 76.98 | 133 | 158 | 18 | 69 | 65.84 | 89.77 | 88.08 | 0.78 | 75.35 |
| SGD + Mom. | 0.1 | 58.73 | 69 | 153 | 23 | 13 | 34.16 | 86.93 | 75.00 | 0.61 | 46.94 |
| | 0.01 | 94.70 | 194 | 164 | 12 | 8 | 96.04 | 93.18 | 94.17 | 0.95 | 95.10 |
| | 0.001 | 95.77 | 195 | 167 | 9 | 7 | 96.53 | 94.89 | 95.59 | 0.96 | 96.06 |
| | 0.0001 | 94.97 | 187 | 172 | 4 | 15 | 92.57 | 94.89 | 97.91 | 0.95 | 95.17 |
| | 0.0000 | 95.77 | 194 | 168 | 8 | 8 | 96.04 | 95.45 | 96.04 | 0.95 | 96.04 |

Table 6-13 Results obtained with the use of regularization L^2 .

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|------------|--------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.1 | 46.56 | 0 | 176 | 0 | 20 | 0 | 100 | 0 | 0.50 | 0 |
| | 0.01 | 84.49 | 153 | 168 | 8 | 49 | 75.74 | 95.45 | 95.03 | 0.86 | 84.30 |
| | 0.001 | 96.03 | 193 | 170 | 6 | 9 | 95.54 | 96.59 | 96.98 | 0.96 | 96.26 |
| | 0.0001 | 94.44 | 193 | 164 | 12 | 9 | 95.54 | 96.59 | 94.15 | 0.94 | 94.84 |
| | 0.0000 | 93.92 | 188 | 167 | 9 | 14 | 93.07 | 94.89 | 95.43 | 0.94 | 93.07 |
| SGD | 0.1 | 66.67 | 180 | 72 | 104 | 22 | 89.11 | 40.91 | 63.38 | 0.65 | 74.07 |
| | 0.01 | 95.50 | 190 | 171 | 5 | 12 | 94.06 | 97.16 | 97.44 | 0.96 | 95.72 |
| | 0.001 | 96.30 | 194 | 170 | 6 | 8 | 96.04 | 96.59 | 97.00 | 0.96 | 96.52 |
| | 0.0001 | 94.44 | 192 | 165 | 11 | 10 | 95.05 | 93.75 | 94.58 | 0.94 | 94.81 |
| | 0.0000 | 71.43 | 117 | 153 | 23 | 85 | 57.92 | 88.93 | 83.57 | 0.72 | 68.42 |
| SGD + Mom. | 0.1 | 65.87 | 134 | 115 | 61 | 68 | 66.34 | 65.34 | 68.72 | 0.66 | 67.51 |
| | 0.01 | 97.09 | 196 | 171 | 5 | 6 | 97.03 | 97.16 | 97.51 | 0.97 | 97.27 |
| | 0.001 | 94.44 | 187 | 170 | 6 | 15 | 92.57 | 96.59 | 96.89 | 0.95 | 94.68 |
| | 0.0001 | 95.50 | 194 | 167 | 9 | 8 | 96.04 | 94.89 | 95.57 | 0.95 | 95.80 |
| | 0.0000 | 96.03 | 193 | 170 | 6 | 9 | 95.54 | 96.59 | 96.98 | 0.96 | 96.26 |

From the results obtained in the table 6-12 and 6-13, it can be observed that although the use of the regularizer did not improve significantly the results, there were some cases in which they improved. For the learning rates of 0.1, 0.01, 0.0001 and 0.00001, using regularizer, and in the SGD optimization function with moment, an improvement in the classifier learning can be verified.

The optimization function with better performance in training without regularizer was the ADAM, and as was said earlier, the optimization function that best suited the training with regularization was the SGD with momentum.

The best overall result for Net2 was obtained with the use of regularizer for a learning rate of LR = 0.01 and SGD optimization function, with the lowest number of false negatives detected (FN = 6), with a low number of false positives (FP = 5), and therefore a better F_1 -score (F_1 -score = 97.27%), sensitivity (SN = 97.03%) and AUC = 0.97. Table 6-14 shows the confusion matrix and figure 6- 8 the respective ROC curve and the AUC value for the best obtained model.

Table 6-14 Confusion matrix for the best focused images classifier, with LR=0.01, SGD optimization function and with L^2 .

| | | Predicted Class | |
|--------------|---------|-----------------|---------|
| | | Focused | Blurred |
| Actual Class | Focused | 171 | 5 |
| | Blurred | 6 | 196 |

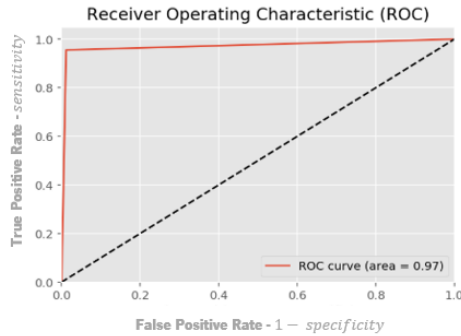


Figure 6-8 ROC curve for the best classification model, with LR=0.01 and SGD optimization function with L^2 . The ROC curve was generated with 3 points.

The classifier was also evaluated by the images that were incorrectly classified, where focused images (true class negative) was classified as blurred (positive class) by the network, and the blurred (positive class) images were classified as focused (negative class) by the network. These images are in Figure 6-9 and 6-10.

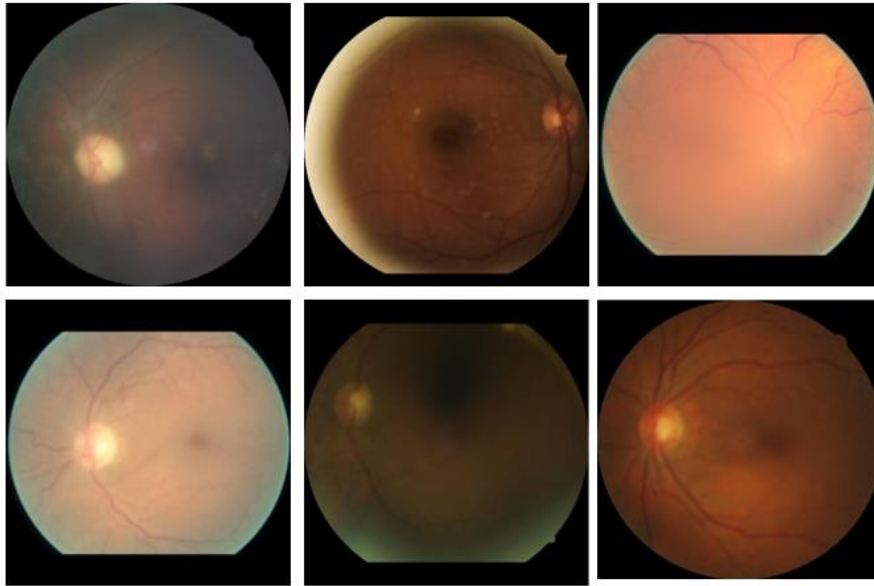


Figure 6-9 Images belonging to the positive class (blurred images) that were classified as focused. In this case none of the images was correctly classified nor surpassed the human classification, since none of the images is focused.

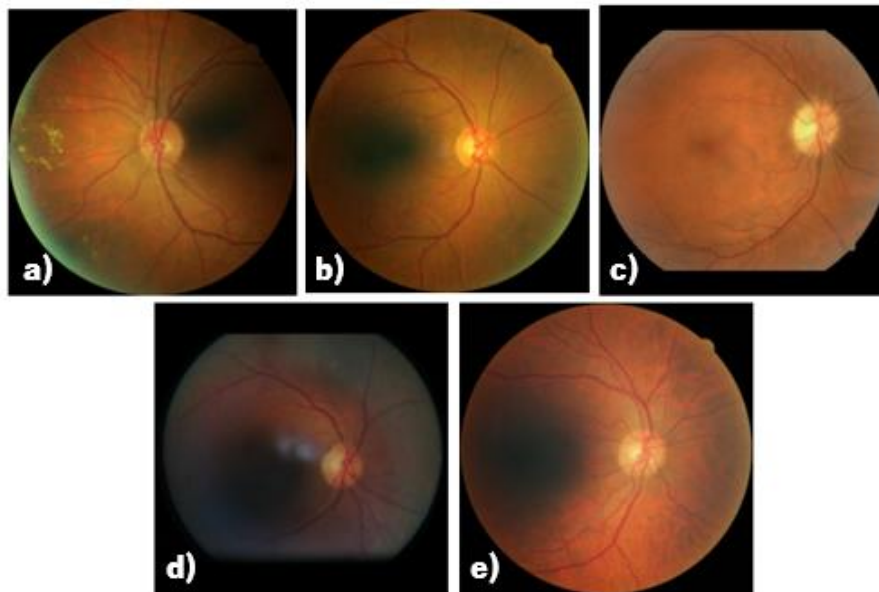


Figure 6-10 Images belonging to the negative class (focused images) that were classified as blurred. Image a), b) and c) are focused while d) and e) were correctly classified by the network and incorrectly classified by the specialist.

After verifying and correcting the labels that were incorrectly classified by the specialist and were correctly classified by the Net2 network and exceeding the human classification, all the metrics involved in the classification were recalculated, obtaining the values of table 6-15. Case A corresponds to the values before the recalculation and in case B are those that were recalculated.

Table 6-15 Comparison of results before and after recalculation of classification metrics.

| Case | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|----------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| A | 97.09 | 196 | 171 | 5 | 6 | 97.03 | 97.16 | 97.51 | 0.97 | 97.27 |
| B | 97.62 | 196 | 173 | 3 | 6 | 97.03 | 98.30 | 98.49 | 0.98 | 97.75 |

The precision in case B has improved after the recalculation of the classification metrics, since the number of FP is much smaller after the correction of classes 0 and 1 of the test images, of the previously incorrect classifications. The other metrics such as F_1 -score, AUC, specificity also improved.

Figure 6-11 shows the information and features learned from the network in each of the five convolutions, represented as Conv2 1, Conv2D #2, Conv2D #3, Conv2D #4 and Conv2D #5.

The same Net1 feature extraction test image was used for the Net2 network.

In the Conv2D layer #1, 11 feature maps were generated that extracted geometric information, the border between the retina and the background, with differentiation of the circle of the retina, the macula and the optical disc; this differentiation can be seen through the different shades of color in feature maps. In addition to all the features extracted above, it's possible to see the segmentation of the blurred zone from the focused area of the image (blurred has a bluish tint and focus has an orange tint) and also the visualization of the surrounding area of the fovea.

From the second convolution layer (Conv2D #2) to the fourth (Conv2D #4) more detailed information on the location, shape and volume of vessels and retinal constituents such as the macula, optic disc and fovea were extracted, as well as a differentiation of the areas out of focus. From the fifth convolution layer (Conv2D #5), feature maps were generated with the targeting of the unfocused zone to dark blue and the zone focused to lighter blue.

The computation time to train and extract features was between 17min30sec and 1h40min4sec, depending on the optimizer and learning rate used. To predict the classes of 378 images, the execution time was 1sec3msec.

The model that took more time to train, had a training time of 1h40min and each training was around to be trained between 40 and 336 epochs.

Globally, the two Net1 and Net2 networks were able to successfully classify and extract features, but Net1 has performed better with 98.68% of test accuracy, 99% AUC and F_1 - score =98.75%.

RESULTS AND DISCUSSION

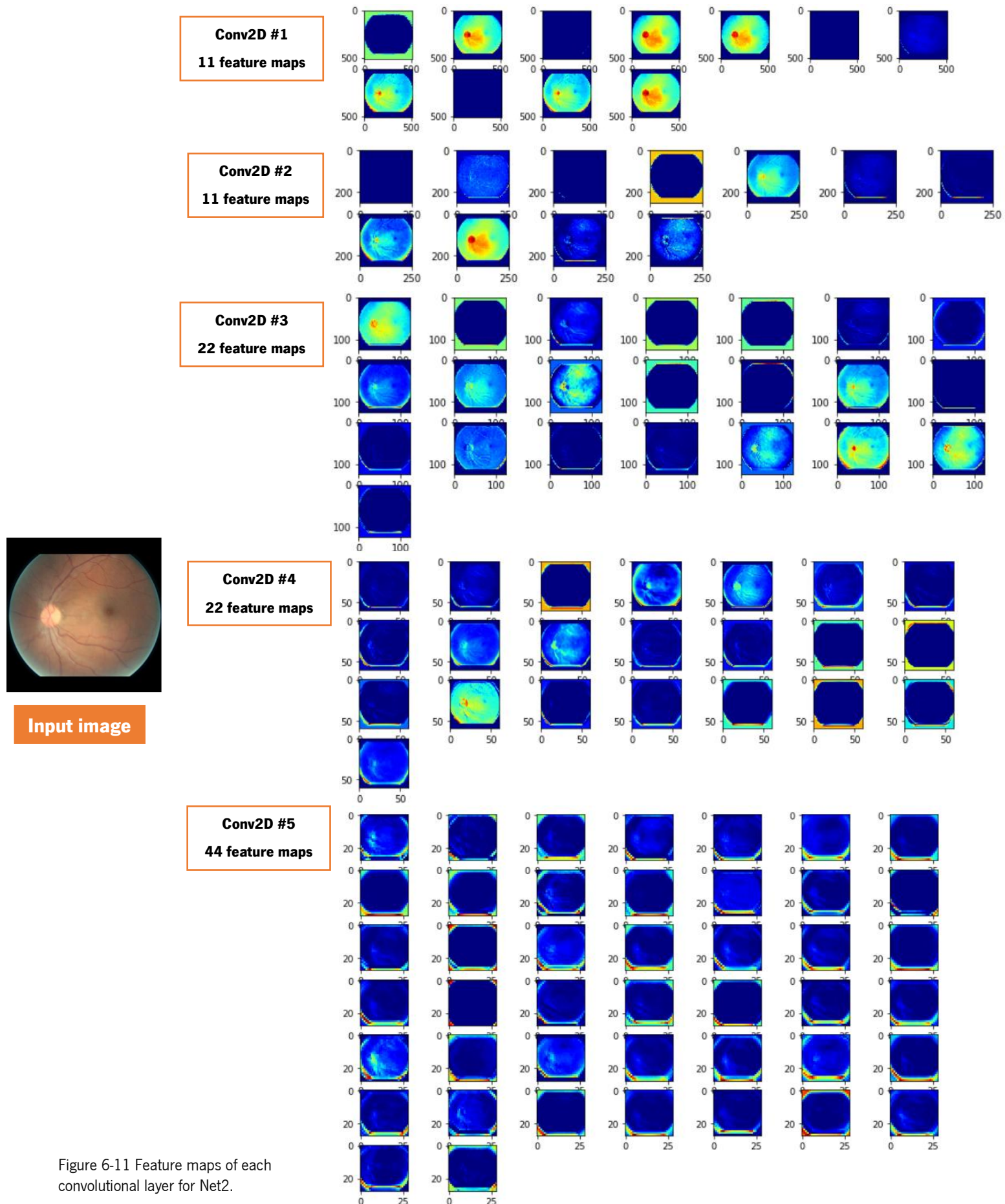


Figure 6-11 Feature maps of each convolutional layer for Net2.

6.3 CLASSIFICATION RESULTS OF COLOR ASSESSMENT

The results of the color parameter are divided between the results obtained in the Net1 and Net2 networks.

To obtain the performance results of the Net1 and Net2 networks, 201 images were used as test dataset, divided into 86 images of class 0 (normal color images), 44 images of class 1 (light color images) and 71 images of class 2 (dark color images). For training and validation, 641 images and 162 images were used respectively. The distribution of the images by training, validation and test dataset can be found in Appendix B-2.

The loss function used was the categorical cross entropy, widely used in multi-class classification, and three different optimization functions were used: Stochastic gradient descent (SGD), Stochastic gradient descent with momentum (SGD + Momentum) and Adam.

As in the focus classification, the color classification was used for dropout regularization with a probability of 0.5 entering the first and second fully connected layers and the L^2 regularization with factor of $\lambda = 0.001$. The fixed and varied parameters used in the training of the two models are presented in table 6-16 and 6-17. As there is some class imbalance in the color dataset, both in training and in validation, which can be seen in Appendix B-2, the micro-average F_1 -score metric is the same value calculated for the test accuracy.

Table 6-16 Fixed parameters used in each model Net1 and Net2.

| Fixed parameter | Value |
|----------------------|---|
| Epochs | 400 epochs (Early Stopping, patience=50 epochs) |
| Convolution Layers | 5 |
| Loss Function | categorical cross entropy |
| Activation function | ReLU (Convolution layers) and Softmax (FCL) |
| Dropout | 0.5 |
| L^2 regularization | 0.001 |
| Momentum | 0.9 |

Table 6-17 Varied parameters used in each model Net1 and Net2.

| Varied parameter | Value |
|------------------------|------------------------------|
| Batch size | 4, 12, 24, 32, 64 |
| Learning rate (LR) | 0.01, 0.001, 0.0001, 0.00001 |
| Optimization Functions | SGD, SGD + Momentum, Adam |

6.3.1 RESULTS OF MODEL NET1

In order to classify color images of the retina, transfer learning of the varied and fixed parameters of the best obtained with the Net1 network in the focus images was used, because they served as a guide for the training to be performed with the color dataset. Therefore, instead of performing the entire pipeline of the 15 trainings performed for each learning rate and each optimization function with and without Batch Normalization, as the focus assessment, only the best classification results were chosen, above the 96.03% test accuracy. Table 6-18 summarizes the classification results.

Table 6-18 Results obtained from the classification of color images of Net1 network.

| Optm. | LR | F_1 -score |
|----------------|---------------|--------------|
| ADAM | 0.001 | 88.06 |
| | 0.0001 | 93.03 |
| SGD | 0.01 | 91.04 |
| | 0.001 | 92.04 |
| SGD + Momentum | 0.01 | 86.07 |
| | 0.001 | 89.55 |
| | 0.0001 | 93.53 |
| | 0.00001 | 91.04 |

After table 6-18 reading, it is verified that the best F_1 -score result (F_1 -score = 93.53%) was obtained with the SGD optimization function with moment, with the learning rate equal to 0.0001 and batch size of 4.

To reduce the slight overfitting, improve network performance and increase the F_1 -score value in the test images, the best model with L^2 regularization was trained and different batch sizes of 12, 24, 32 and 64 were used. The results are shown in table 6-19. And the learning of the network is shown in figure 6-12.

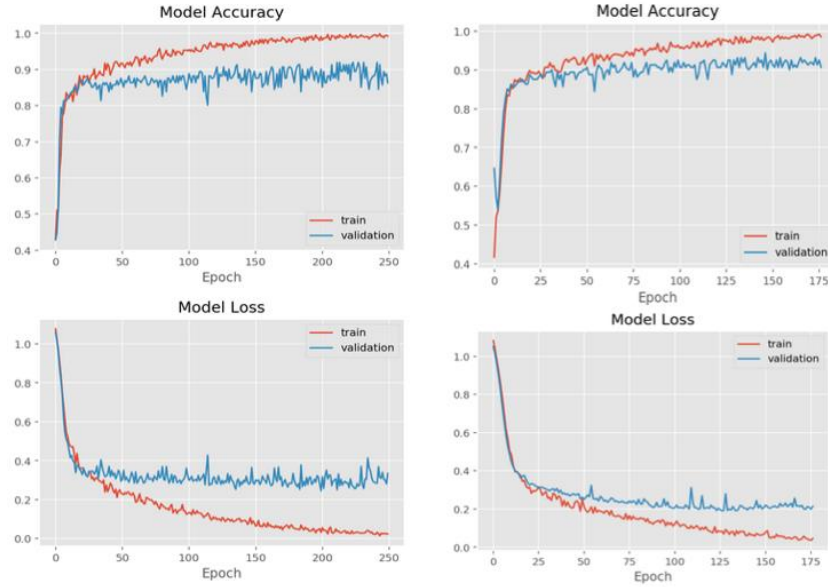


Figure 6-12 Learning curves for the model trained with Batch Normalization and without regularization L^2 (left) and with regularization L^2 (right).

There was learning improvement in use regularization in the model training, with reduction of overfitting and therefore, obtaining better values of training and validation accuracy and reduction of validation loss. The use of regularization stabilized, accelerating the training process, since there is less variation and oscillation in the learning curve of the training loss and training accuracy, as well as fewer number of iterations/epochs for the model to be trained (without regularization = 250 epochs and with regularization = 177 epochs).

Table 6-19 Results of the classifier performance for LR = 0.0001 and SGD optimization function with momentum, using Batch Normalization, different batch sizes and the use of L^2 regularization.

| Use of regularization | Batch size | F_1 -score |
|-----------------------|------------|--------------|
| No | 4 | 93.53 |
| Yes | 4 | 94.53 |
| No | 12 | 90.55 |
| No | 24 | 91.04 |
| No | 32 | 91.04 |
| No | 64 | 90.55 |

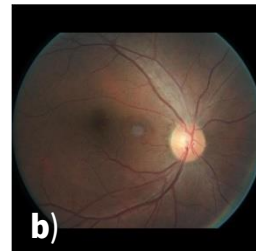
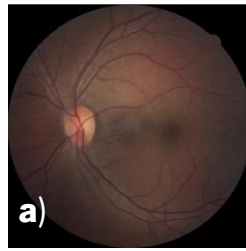
What can be concluded from the results of Table 6-19 is that the use of regularization improved the value of F_1 -score, contrary to the increase of batch size. In the table 6-20, the confusion matrix for the best classifier obtained by Net1 is shown, and what can be verified, the class that was mostly badly classified was the “bright” class (class 1), with 5 wrong classifications in 41 images. None of the images that were of class “normal” were incorrectly classified as “bright”, that is, it can be verified that the classifier was able to extract and differentiate the normal color characteristic of bright color but not vice versa, obtaining 2 wrong classifications of “bright” color (true class) to “normal” (predicted class). Finally, the class that suffered less with the incorrect classifications was the “normal” class (class 0), having only 2 incorrect classifications in 86 images. In a total of 201 images, 11 images were classified differently by the human and the network.

Table 6-20 Confusion matrix for the best result obtained from Net1 network, with Batch Normalization and L^2 regularization.

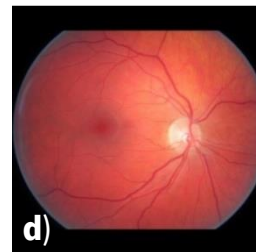
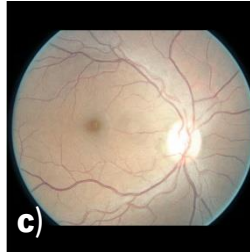
| | | Predicted Class | | |
|--------------|--------|-----------------|--------|------|
| | | Normal | Bright | Dark |
| Actual Class | Normal | 84 | 0 | 2 |
| | Bright | 2 | 39 | 3 |
| | Dark | 3 | 1 | 67 |

In Figure 6-13, the images are classified by Net1 as a given class and belonging to another that was obtained, in the process of data acquisition, through human classification. Images a) and b) belong to the class “normal” and were classified as “dark” by the network; Images c) and d) are of the “bright” class and are predicted to be of the “normal” class. Images (e) to g) are classified by the human as “bright” the images are of the “dark” class and predicted by the network as of the “normal” class. Finally, the image k) is classified as “dark” but predicted as “bright”.

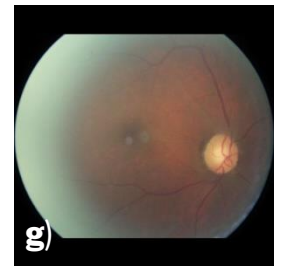
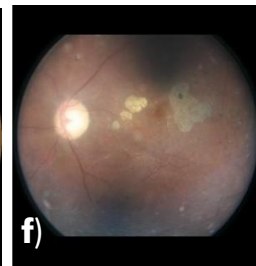
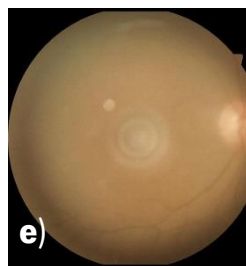
Normal → Dark



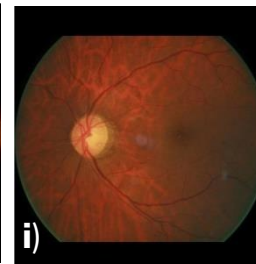
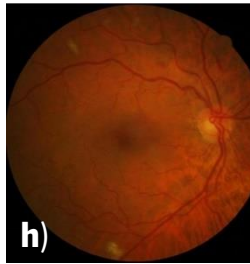
Bright → Normal



Bright → Dark



Dark → Normal



Dark → Bright

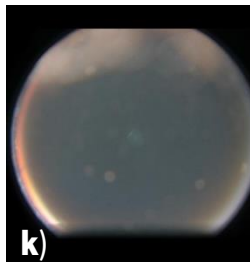


Figure 6-13 Images that the network classified differently from the human.

Of all the images differently classified between the human and the network (11 images), 5 were correctly classified by the network and surpassed the human classification, which shows that the network generalized well and was able to surpass the human classification.

The images a) and b) are definitely from class “dark”, d) it is “normal”, h) is “normal” despite being a bit darker as usual. The classification of image k) is divided between the human and the network since it contains both dark and bright characteristic, so, on this image, one can conclude that it is on the threshold of the two characteristics and both classifications (human and network)

are also accepted. Due to some test images being misclassified by the human and correctly classified by the Net1 network, F_1 -score classification value was recalculated from the corrected values of TP, TN, FP and FN, shown in table 6-21. Case A corresponds to the case before the recalculation of the classification metric F_1 -score and case B is the one in which it has already been recalculated. From table 6-21, it can be seen that the value of F_1 -score in B increased by 2.48% in relation to case A.

Table 6-21 Comparison of results before and after recalculation of classification metrics.

| Case | F_1 -score |
|----------|--------------|
| A | 94.53 |
| B | 97.01 |

In order to understand how the network interpreted the color images, feature maps were generated of a test image of the “bright” class, and this representation is shown in figure 6-14.

In the first convolution layers (Conv #1 and Conv #2), the geometric features of the border were extracted, but also the internal organization of the retina, with macula, optic disc and fovea differentiations. Still from these layers were extracted the different shades and the brightness feature of this image. In the following layers, Conv #3 to Conv #5, the optic disc is more evidenced and the region with the highest brightness of the image also (with yellowish and orange tint), or darker as appear in the third feature map of the Conv #3 and second feature map of Conv #4.

As can be concluded, all the important features to be extracted from the “bright” class image were successfully extracted and learned by the network.

The nets were trained between 5min13sec and 21min33sec, that is, these train weren't time expensive, since they trained fast, with a batch size of 4 and with 400 epochs and Early Stopping. The training with the shortest time spent (5min13sec), trained with 70 epochs/iterations and the training that took more (21min33sec) trained until 260 epochs, which shows that it was not necessary more than this number of epochs to converge and find a low value of loss and a high value of validation accuracy, as shown in figure 6-12. The execution time and prediction of classes was between 1sec5msec and 2sec11ms. This time difference is due to the fact that the available memory in the training machine, at the time of predictions, differ, interfering in the loading time of the test images and the print output of the network.

RESULTS AND DISCUSSION

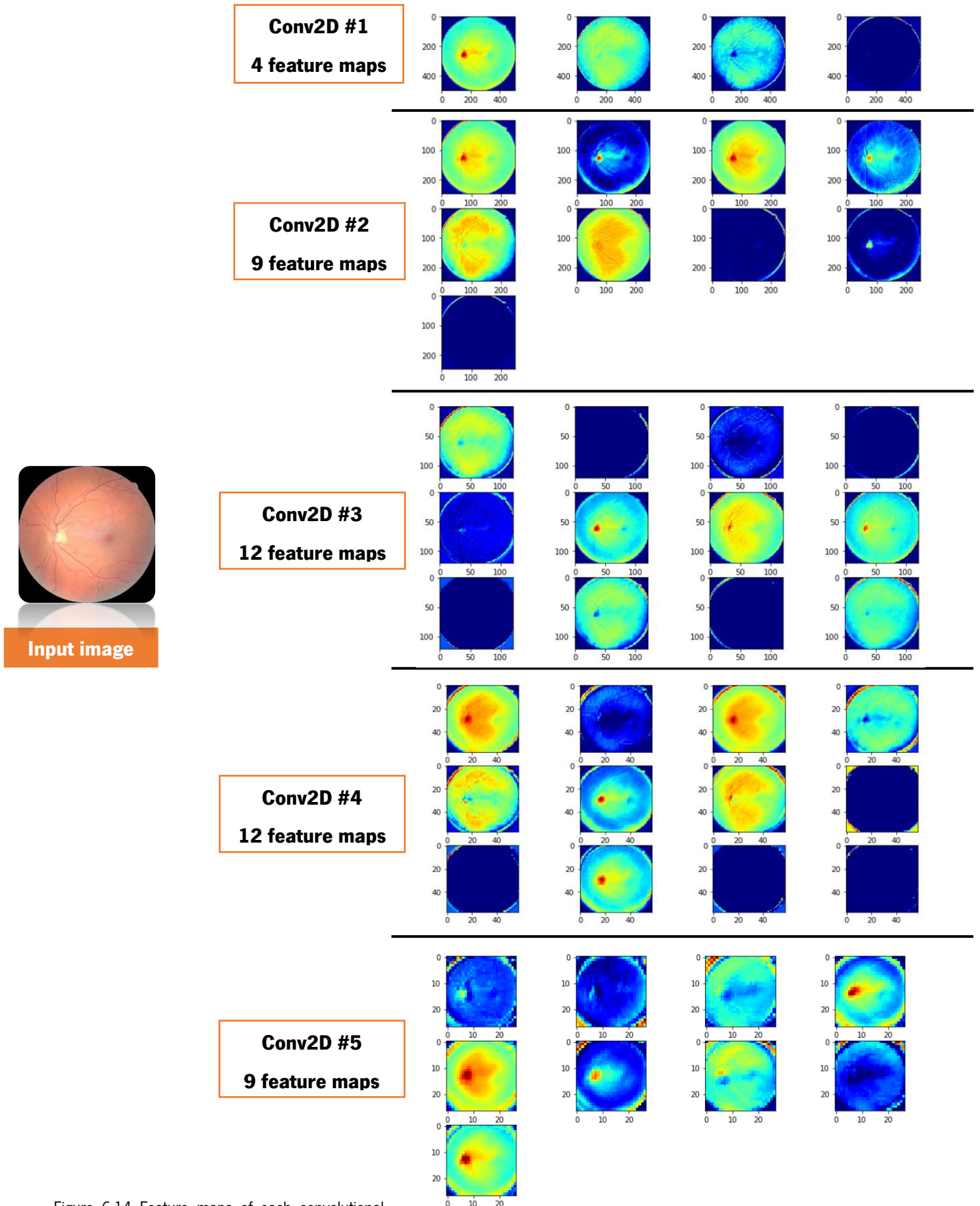


Figure 6-14 Feature maps of each convolutional layer for Net1, of a bright test image.

6.3.2 RESULTS OF MODEL NET2

As in Net1 network, for the Net2 network, transfer learning of the varied and fixed parameters of the best models obtained with the Net2 network in the focus images, were used. All the training pipeline was performed with Batch Normalization and the parameters of the eight best models trained in the focus images were chosen, that is, above the 94.97% test accuracy of the Net2 network. Table 6-22 summarizes the results of the classification.

Some overfitting occurred, for example, for LR = 0.0001 and ADAM optimization function with Batch Normalization. To overcome this phenomenon, L^2 regularization was used, to reduce or mitigate the overfitting effect, and by figure 6-15, it's verified that it reduced the overfitting, stabilized the learning, increased the accuracy of training and validation, as well as reduce the training loss and validation, converging the latter to a low value and close to zero - there was improvement in learning performance. The regularization implementation also increased the F_1 -score, going from 92.54% without regularizer, to 94.53% with regularizer.

It was used regularizer for the three LR (0.01, 0.001 and 0.0001) and ADAM optimization function and the results of these trainings are presented in table 6-23.

Table 6-22 Results of the color image classification of Net2.

| Optm. | LR | F_1 -score |
|----------------|---------------|--------------|
| ADAM | 0.01 | 89.05 |
| | 0.001 | 92.54 |
| | 0.0001 | 92.54 |
| SGD | 0.001 | 89.55 |
| | 0.0001 | 90.55 |
| SGD + Momentum | 0.001 | 90.55 |
| | 0.0001 | 91.54 |
| | 0.00001 | 88.06 |

Table 6-23 Results of images classification, with the L^2 regularization implementation, in Net2.

| Optm. | LR | F_1 -score |
|-------|---------------|--------------|
| ADAM | 0.01 | 90.05 |
| | 0.001 | 93.53 |
| | 0.0001 | 94.53 |

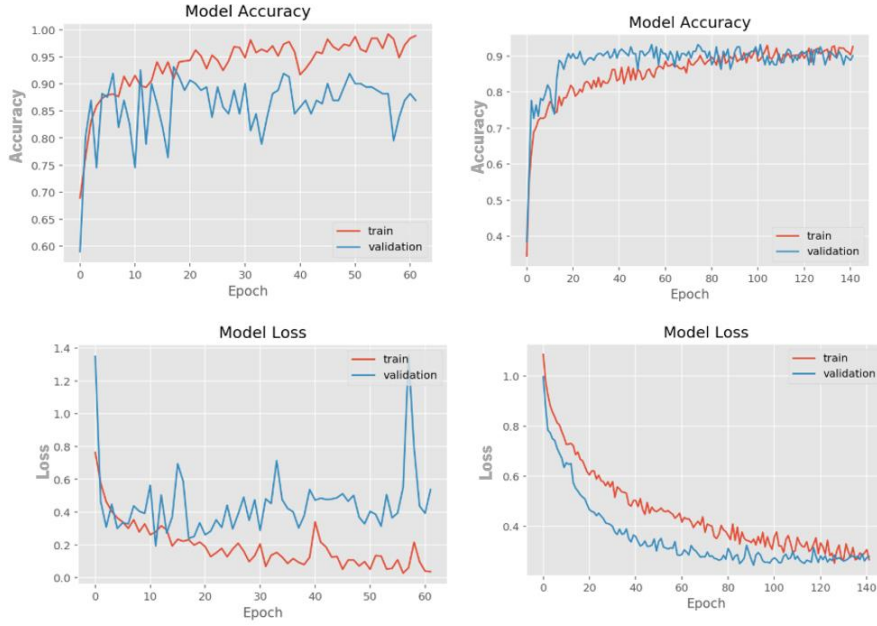


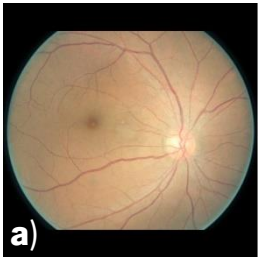
Figure 6-15 Accuracy and Loss learning curves along the training, for the train and validation datasets, without regularization on the left and with regularization on the right.

After reading Table 6-23, it can be seen that all previous values of F_1 -score obtained only with Batch Normalization, increased with the use of L^2 regularization. The best F_1 -score result (F_1 -score = 94.53%) was obtained with the Adam optimization function, with the learning rate of 0.0001 and batch size of 4. The following confusion matrix, represented in table 6-24, corresponds to the best model of the Net2 network, obtained with regularization, being able to correctly identify 81 images of the class “normal” in a total of 86, 39 images of class “bright” in 44 and 70 images of class “dark” in 71 images. The “dark” class was the class with the lowest number of incorrect classifications given by the network, with a total number of 1 image classified as “normal” color. The network predicted 11 images, contrary to the human, in a total of 201 images. Table 6-24 represents the confusion matrix with the correct and incorrect predicted classes.

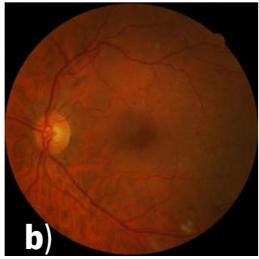
Table 6-24 Confusion matrix with the distribution of the classification given by Net2 network.

| | | Predicted Class | | |
|--------------|--------|-----------------|--------|------|
| | | Normal | Bright | Dark |
| Actual Class | Normal | 81 | 1 | 4 |
| | Bright | 4 | 39 | 1 |
| | Dark | 1 | 0 | 70 |

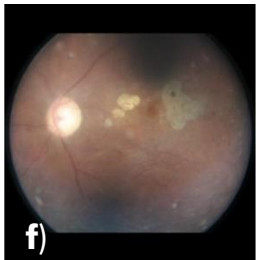
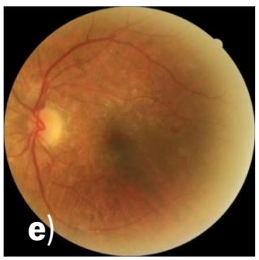
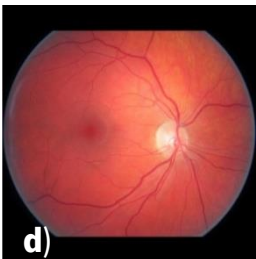
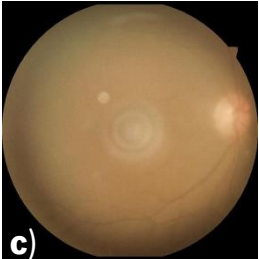
Normal → Bright



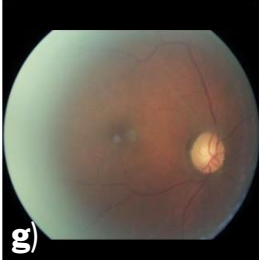
Dark → Normal



Bright → Normal



Bright → Dark



Normal → Dark

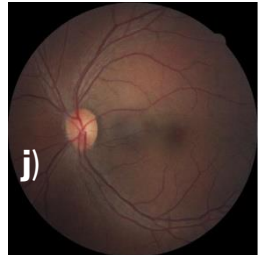
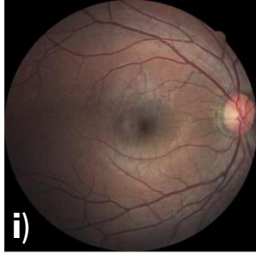
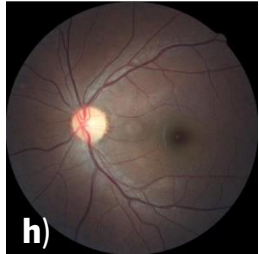


Figure 6-16 Images with classification made by Net2 network, different from human classification.

RESULTS AND DISCUSSION

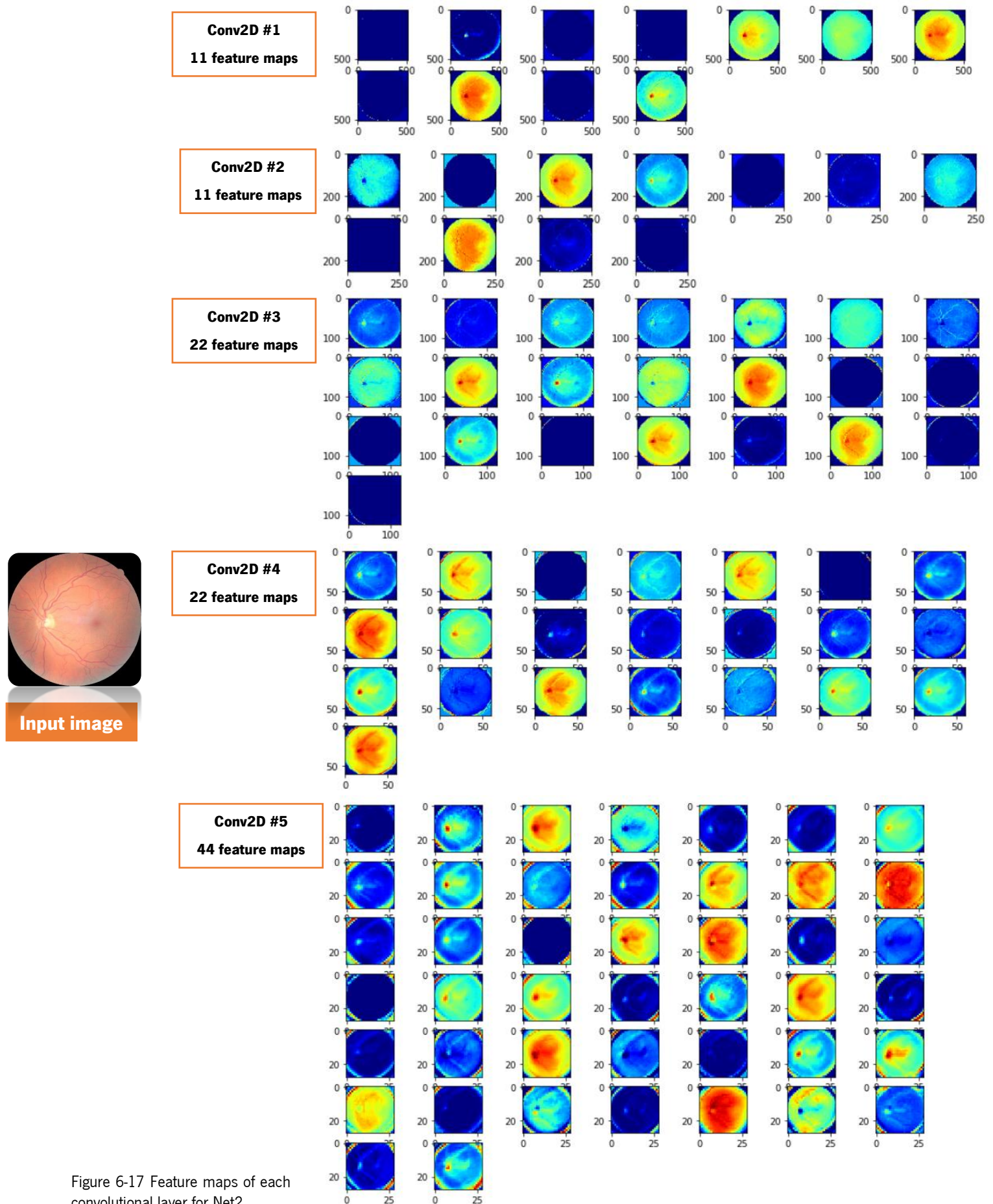


Figure 6-17 Feature maps of each convolutional layer for Net2.

In Figure 6-16, from a) to k) are the images that were differently classified from the human classification. The images a), b), c), e), f), j) and k) were incorrectly classified by the network, but images d), h), i) were correctly classified by the network, where d) belongs is truly from class “normal” and h), i) and j) belong to the “dark” class. In 11 images, 4 were classified correctly by the network and surpassed the selection of the images made by the human.

F_1 -score metric was recalculated, since the network classified some images correctly, which the expert had not previously done, obtaining a higher value (F_1 -score = 96.52%) than previously predicted (still with the incorrect human classification, F_1 -score = 94.53%). These results are shown in Table 6-25, where case A is before the F_1 -score recalculation and B is after the recalculation.

Table 6-25 Comparison of results before and after recalculation of classification metrics.

| Case | F_1 -score |
|----------|--------------|
| A | 94.53 |
| B | 96.52 |

Figure 6-17 shows the feature maps extracted from the convolution layers of Net2 network of a bright image. From the first layer (Conv2D #1) to the last layer (Conv2D #5) it can be seen that the features of location and retinal constituents, their border with the background, and a light color segmentation with respect to the remaining color of the retina (the light color is represented by a more orange and yellowish tint in the feature maps). From these feature maps, the shape and structure of retinal vessels are also evident.

The training time of the studied networks was between 4min49sec and 19min44sec, taking less time than the Net1 training network. The training with 4min49sec required 91 epochs and the training with 19min44sec, 159 epochs. Figure 6-15 (right side), for example has a training time of 10min50sec and required 142 epochs of 400 using Early Stopping. The use of Early Stopping causes the training to progress until the validation loss reaches a low value and stabilizes. Class prediction and classification time was between 1sec4ms and 4sec19msec. This difference, as with the Net1 network, is due to the fact that the available memory oscillates and differs at the time of the classes prediction.

The study and evaluation of the color parameter, made by the network Net1 and Net2, was completed successfully, since the values of F_1 -score were high, and equal to 97.01% for Net1 and 96.52% for Net2.

One of the major limitations in the study of color was the fact that the human classification failed to correctly distribute the test images, to their correct classes, which created some margin of error in the prediction of the classes, as can be seen from the analysis of the confusion matrices of both Net1 and Net2. On the other hand, these images poorly classified by the human were, however, correctly classified by the network, surpassing the human classification. The net that most features learned, and the failed least was network Net1, having from 11 images, 5 images correctly classified by him and therefore 97.01% of F_1 -score.

In general, the two Net1 and Net2 networks have extracted the most important features and information to distinguish a “normal” image, an image with “bright” color and a “dark” image and this can be seen by the feature maps extracted from the convolution layers, as well as the F_1 -score great values.

6.4 CLASSIFICATION RESULTS OF ILLUMINATION ASSESSMENT

The results of the illumination parameter (classes “even” and “uneven”) are divided between the obtained results in Net1 and Net2 networks.

To test the performance of Net1 and Net2, 320 images were used, divided into 180 images of class 0 (images of even illumination) and 160 images of class 1 (images of uneven illumination). For training and validation, 1610 images and 500 images were used respectively. The distribution of the images by training, validation and test datasets are given in Appendix B-3.

The loss function used was binary cross entropy, since a binary classification was used, and three different optimization functions were used: Stochastic gradient descent (SGD), Stochastic gradient descent with momentum (SGD + Momentum) and Adam.

As in the classification of focus and color parameters, for the illumination classification, were used dropout regularization with probability of 0.5 between the first and second fully connected layers and regularization L^2 with factor of $\lambda = 0.001$.

The fixed and varied parameters used in the training of the two models are presented in tables 6-26 and 6-27.

Table 6-26 Fixed parameters used in each model Net1 and Net2.

| Fixed parameter | Value |
|----------------------|---|
| Epochs | 400 epochs (Early Stopping, patience=50 epochs) |
| Batch size | 4 |
| Convolution Layers | 5 |
| Loss Function | binary cross entropy |
| Activation function | ReLU (Convolution layers) and Sigmoid (FCL) |
| Dropout | 0.5 |
| L^2 regularization | 0.001 |
| Momentum | 0.9 |

Table 6-27 Varied parameters used in each model Net1 and Net2.

| Varied parameter | Value |
|------------------------|------------------------------|
| Learning rate (LR) | 0.01, 0.001, 0.0001, 0.00001 |
| Optimization Functions | SGD, SGD + Momentum, Adam |

6.4.1 RESULTS OF MODEL NET1

In order to classify the images of the retinal illumination parameter, transfer learning of the varied and fixed parameters of the best models obtained with the Net1 network in the focus images, above 96.03% of the test accuracy, were used. Table 6-28 summarizes the results of the classification. All the training pipeline were carried out with Batch Normalization.

Table 6-28 Results obtained from the illumination classification of Net1.

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|------------|---------------|--------------|------------|------------|----------|-----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.001 | 78.75 | 125 | 127 | 33 | 35 | 78.13 | 79.38 | 79.11 | 0.79 | 78.62 |
| | 0.0001 | 93.75 | 149 | 151 | 9 | 11 | 93.13 | 94.38 | 94.30 | 0.94 | 93.71 |
| SGD | 0.01 | 91.25 | 150 | 142 | 18 | 10 | 93.75 | 88.75 | 89.29 | 0.91 | 91.46 |
| | 0.001 | 90.94 | 148 | 143 | 17 | 12 | 92.50 | 89.38 | 89.70 | 0.91 | 91.08 |
| SGD + Mom. | 0.01 | 91.56 | 140 | 153 | 7 | 20 | 87.50 | 95.63 | 95.24 | 0.92 | 91.21 |
| | 0.001 | 91.25 | 146 | 146 | 14 | 14 | 91.25 | 91.25 | 91.25 | 0.91 | 91.25 |
| | 0.0001 | 91.25 | 149 | 143 | 17 | 11 | 93.13 | 89.38 | 89.76 | 0.91 | 91.41 |
| | 0.00001 | 89.06 | 144 | 141 | 19 | 16 | 90.00 | 88.13 | 88.34 | 0.89 | 89.16 |

What can be seen from table 6-28 is that the test accuracy, AUC and F_1 -score values for all constructed models are greater than 90%, except for the models trained with ADAM optimization function and LR = 0.001 and the model trained with the SGD optimization function with momentum and LR=0.00001.

The highest test accuracy, F_1 -score and AUC values, obtained by the Net1 network, are highlighted in the table and obtained with a learning rate of 0.0001 and ADAM optimization function. However, it was not the model with the highest sensitivity and the lowest number of FN, detecting one more FN than the model with the highest sensitivity - model with FN=10, for LR=0.01 and SGD optimization function. This model has also not the lowest FP detection, since it has a rate of 94.30% and not 95.24% as obtained the model trained with the optimization function SGD with momentum and LR=0.01. Overall, the model that performed best in all classification metrics was 93.75% accuracy, 0.94 AUC and a F_1 -score of 93.71%.

In order to reduce overfitting and to increase the classification metric values, in the best model, the use of L^2 regularization was implemented.

Figure 6-18 shows the graphs of the validation and training learning process, without regularization (left) and with regularization (right). By the analysis of the figure, the use of regularization did not reduce the overfitting phenomenon, that had already happened previously in the training without regularization. It would be expected that with the use of regularizer and Batch Normalization, in addition to the training loss, also the validation loss would decrease, and, therefore, learning stabilizes and converges to a low loss value, which was not case.

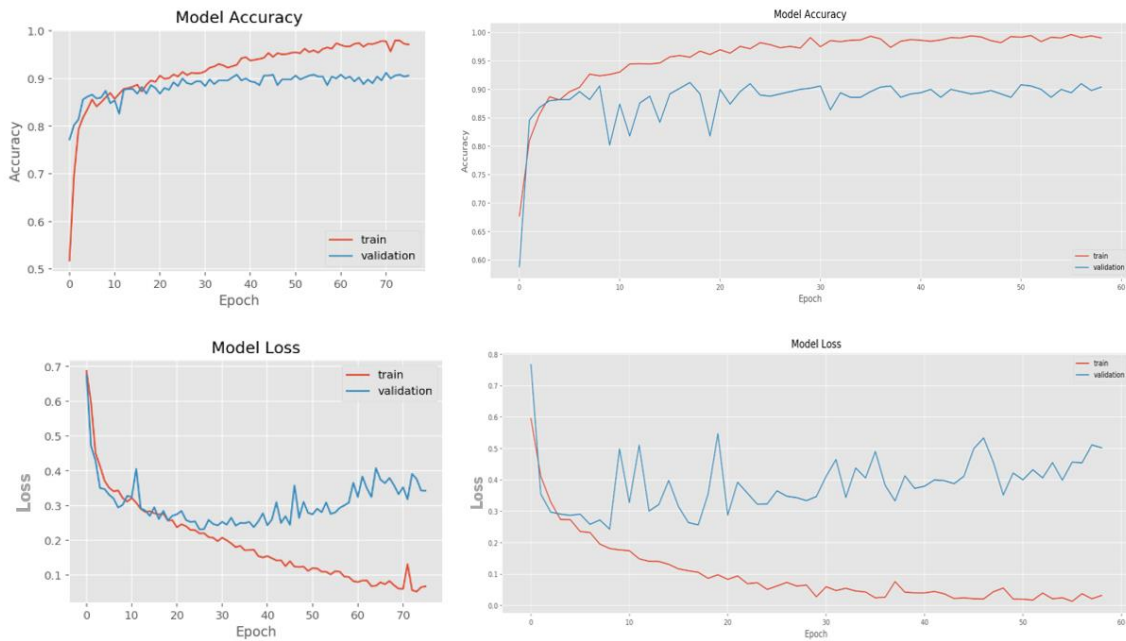


Figure 6-18 Learning curves, of train and validation for the two trained models, only with Batch Normalization (left) and Batch Normalization+ L^2 regularization (right).

Table 6-29 shows a comparison of the classification performance metric values of the models presented in figure 6-18, where it can be verified that, despite the use of regularization, not bring improvements in metric values of test accuracy, specificity, AUC, F1-score and predicting a greater number of false detections (of the positive class - "uneven"), was able to detect smaller number of FN (false detections of the class to be "even" that aren't in fact), and, therefore, obtained greater sensitivity ratio compared to the other model without regularization. The objective of the classifier used is that it provides the lowest number of FN, so the best model was obtained with regularizer. Table 6-30 corresponds to the confusion matrix of the regularization model and Figure 6-19 to the ROC curve graph, with its AUC value, elaborated with 3 points.

Table 6-29 Comparison of the results obtained by the trained model with and without regularization.

| Use of regularization | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|-----------------------|--------------|------------|------------|-----------|----------|--------------|--------------|--------------|-------------|--------------|
| No | 93.75 | 149 | 151 | 9 | 11 | 93.13 | 94.38 | 94.30 | 0.94 | 93.71 |
| Yes | 93.13 | 153 | 145 | 15 | 7 | 95.63 | 90.63 | 91.07 | 0.93 | 93.30 |

Table 6-30 Confusion matrix with the distribution of the classification given by Net1 network.

| | | Predicted Class | |
|--------------|--------|-----------------|--------|
| | | Even | Uneven |
| Actual Class | Even | 145 | 15 |
| | Uneven | 7 | 153 |

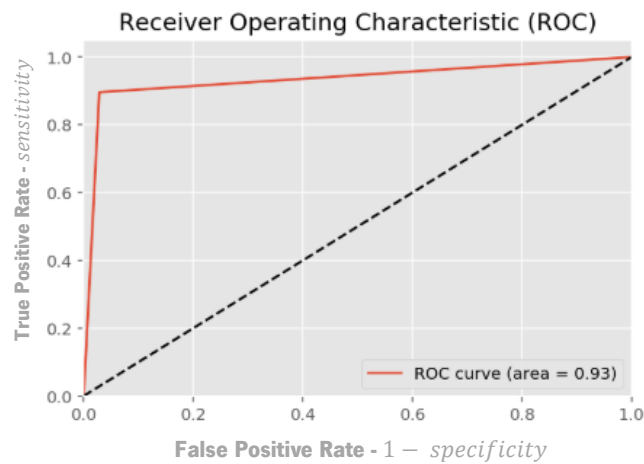


Figure 6-19 ROC curve for the best classification model, with LR=0.0001 and Adam optimization function. ROC curve was created with 3 points.

Figure 6-20 and 6-21 represent the images that Net2 classified differently from the human graders. Supporting the results of the confusion matrix (table 6-30), there are 15 FP and 7 FN. The images present in figure 6-20 were classified by the network as being even and two of them are in fact (2 images in a total of 7 FN). In Figure 6-21, 6 images were correctly classified by the network, totaling 15 FP. It can be concluded that the network has been able to learn the features that differentiate the images with even and uneven illumination and has the power to generalize well for cases not seen before.

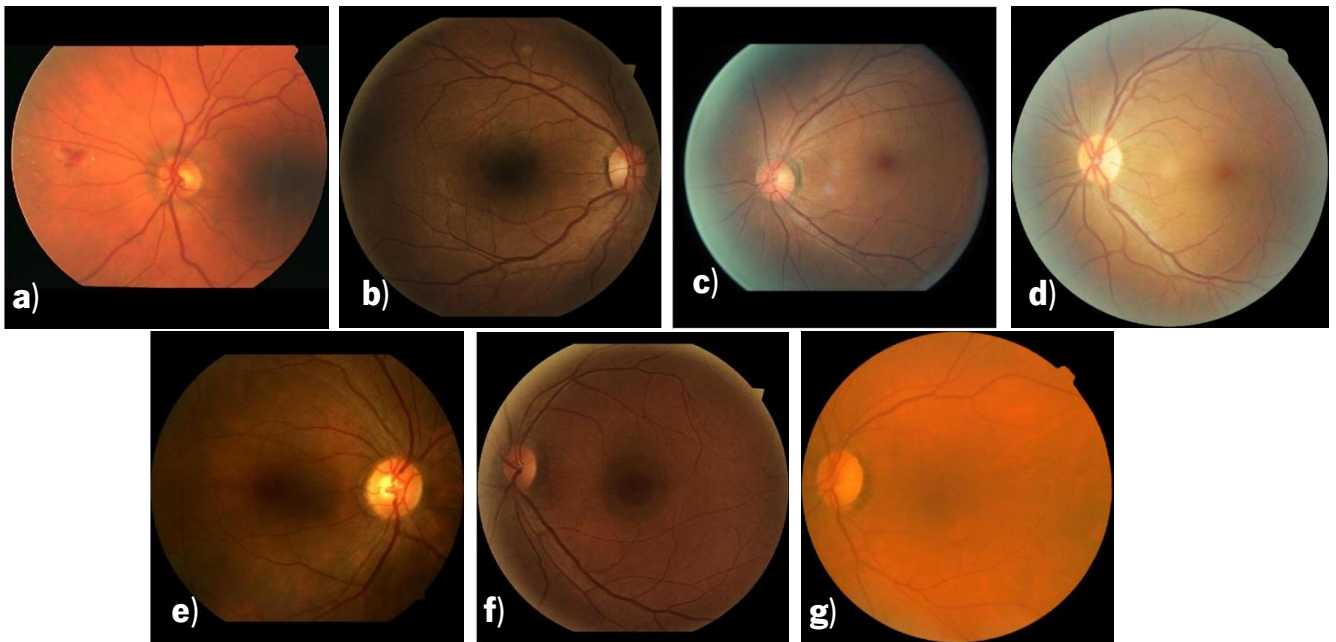


Figure 6-20 Images belonging to the positive class (images classified by the human as uneven illumination) and that were classified as negative class (even) by the network. Images f) and g) were correctly classified by the network since their illumination is regular along all the retinal perimeter. These two cases exceeded the human classification, and the network generalized well for these images.

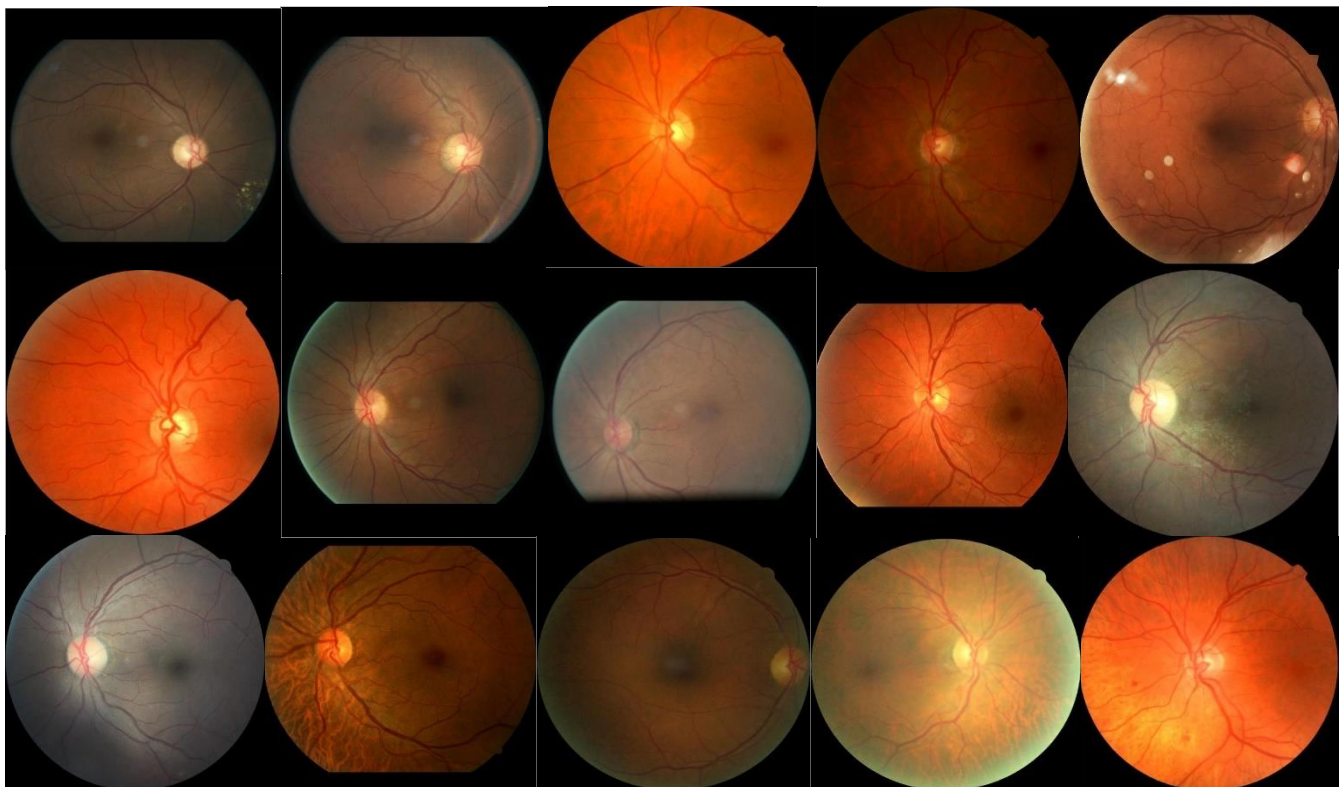


Figure 6-21 Images belonging to the negative class (images classified by the human as even illumination) and that were classified by the network as positive class (uneven). The last image of the second line and all the images that belong to the last line of images are correctly classified as uneven by the network and incorrectly classified by the human classification.

As there were quite a few images correctly classified as class “even” and class “uneven” by the network and surpassed the human classification, it was recalculated the classification metrics, already with this correction. Table 6-31, contains the metric values without class correction in case A and with class correction in case B.

Table 6-31 Comparison of results before and after recalculation of classification metrics.

| Case | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|----------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| A | 93.13 | 153 | 145 | 15 | 7 | 95.63 | 90.63 | 91.07 | 0.93 | 93.30 |
| B | 95.63 | 155 | 151 | 9 | 5 | 96.88 | 94.38 | 94.51 | 0.96 | 95.68 |

The case B, with the recalculation of the metrics and the values TP, TN, FP and FN, exceeded the case A in which it only has a prediction error of 4.37% and smaller than the case A with error equal to 6.87 %.

The generated maps for a “uneven” test image are represented in figure 6-22. It is found that along the convolution layers important features were extracted, such as the dark taint that extends from the macula to the retinal edges, and was confused with the background, represented by dark blue or reddish orange, depending on the feature map. All other features such as the location of the retinal constituents and the retina's borders were also extracted.

All the resulting models were trained between 12min50s and 1h08min35sec, being training sessions that had some time cost associated. These models were trained with a batch size of 4 and with 400 epochs and Early Stopping, which means that until the validation loss curve converges, training continues. The fact that the batch size is also small, makes the model take longer to learn the features present in the input images.

The training with the shortest time (12min50sec) was trained with 72 epochs/iterations and the training that took the longest (1h08min35sec) was finished after 231 epochs.

The runtime and class prediction were for all models equal to 1sec3msec.

RESULTS AND DISCUSSION

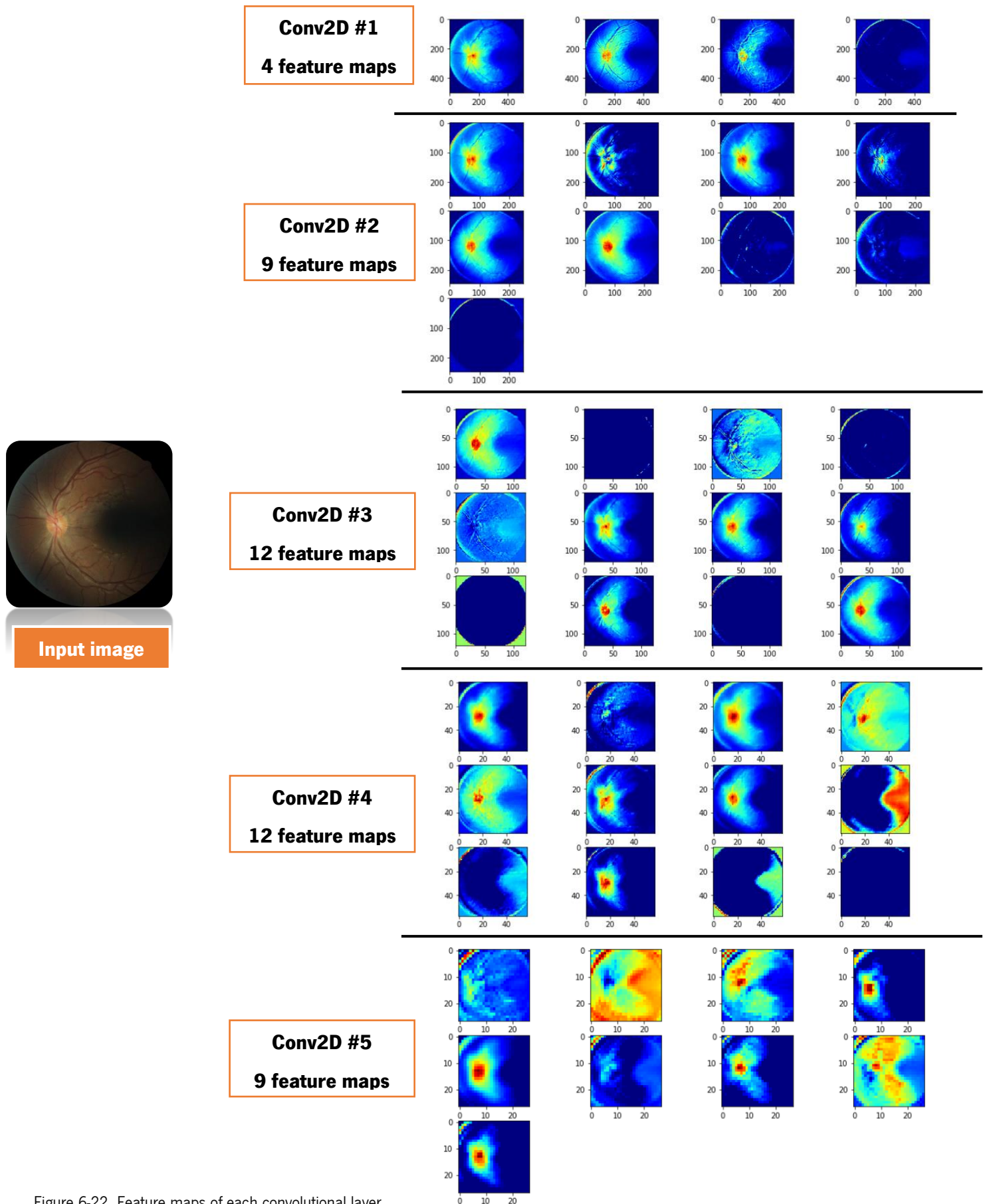


Figure 6-22 Feature maps of each convolutional layer for Net1, of an uneven test image.

6.4.2 RESULTS OF MODEL NET2

As in the Net1 network, for the Net2 network, transfer learning of the varied and fixed parameters of the best models obtained with the Net2 network in the focus images, were used. All training sessions were performed with Batch Normalization and the parameters of the eight best models trained in the focus images were chosen, that is, above the 94.97% test accuracy of the Net2 network. Table 6-32 summarizes the classification results.

Table 6-32 Results obtained from the illumination classification of Net2 network.

| Optm. | LR | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|----------------------|---------------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| ADAM | 0.01 | 87.50 | 136 | 144 | 16 | 24 | 85.00 | 90.00 | 89.47 | 0.88 | 87.18 |
| | 0.001 | 92.19 | 147 | 148 | 12 | 13 | 91.88 | 92.50 | 92.45 | 0.92 | 92.16 |
| | 0.0001 | 94.38 | 150 | 152 | 8 | 10 | 93.75 | 95.00 | 94.94 | 0.94 | 94.34 |
| SGD | 0.001 | 91.25 | 151 | 141 | 19 | 9 | 94.38 | 88.13 | 88.62 | 0.91 | 91.52 |
| | 0.0001 | 92.81 | 153 | 144 | 16 | 7 | 95.63 | 90.00 | 90.53 | 0.93 | 93.01 |
| SGD + Momentum | 0.001 | 91.25 | 148 | 144 | 16 | 12 | 92.50 | 90.00 | 90.24 | 0.91 | 91.52 |
| | 0.0001 | 90.94 | 148 | 143 | 17 | 12 | 92.50 | 89.37 | 89.70 | 0.91 | 91.08 |
| | 0.00001 | 92.81 | 149 | 148 | 12 | 11 | 93.13 | 92.50 | 92.55 | 0.93 | 92.84 |

The model that stood out for the good results in most of the metrics was the model trained with learning rate of 0.0001 and Adam optimization function, obtaining an accuracy of 94.38%, precision of 94.94%, an AUC of 0.94 and an F_1 -score of 94.34 %. Despite these good results, it is not the most sensitive model and, therefore, it detects less FN. This model was trained with LR = 0.0001 and SGD and detected FN = 7, containing a sensitivity of 95.63%. However, the model with the lowest number of false-positive detections of the positive class, detecting a total of 8 images, with a higher value of F_1 -score and accuracy of the entire pipeline of training performed, was the one performed with LR=0.0001 and Adam optimization function. In Table 6-32, the confusion matrix of this last interpreted model is represented and in Figure 6-23 its ROC curve is represented. In Figures 6-24 and 6-25 are the images that the Net2 network classified contrary to the human classification previously made.

Table 6-33 Confusion matrix with the distribution of the classification given by Net2 network, for LR=0.0001 and Adam.

| | | Predicted Class | |
|--------------|--------|-----------------|--------|
| | | Even | Uneven |
| Actual Class | Even | 152 | 8 |
| | Uneven | 10 | 150 |

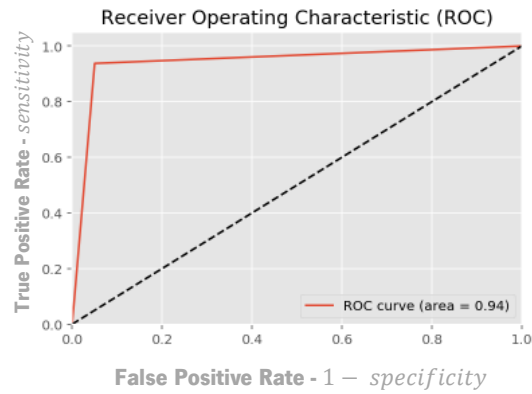


Figure 6-23 ROC curve and AUC value. ROC generated with 3 points.

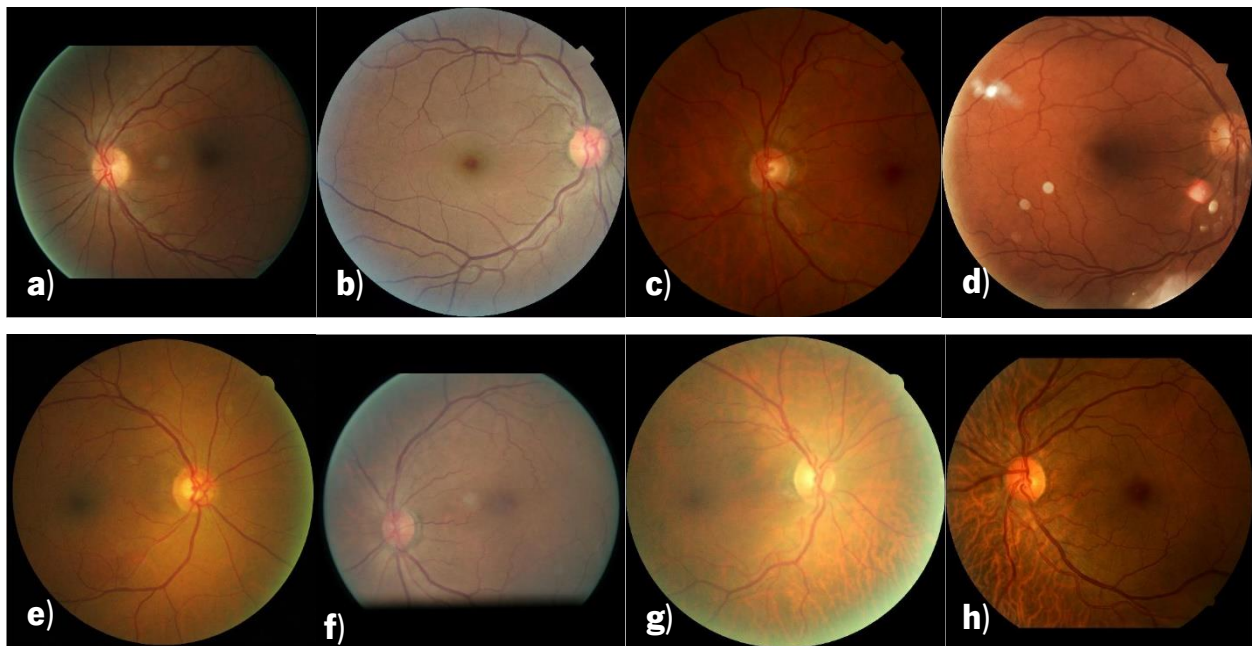


Figure 6-24 Images belonging to the negative class (images classified by the human as even illumination) and were classified by the network as positive class (uneven). Image d) contains artifacts and the network confused as a case of uneven illumination. Images g) and h) were correctly classified by the network as uneven and incorrectly classified by human classification.

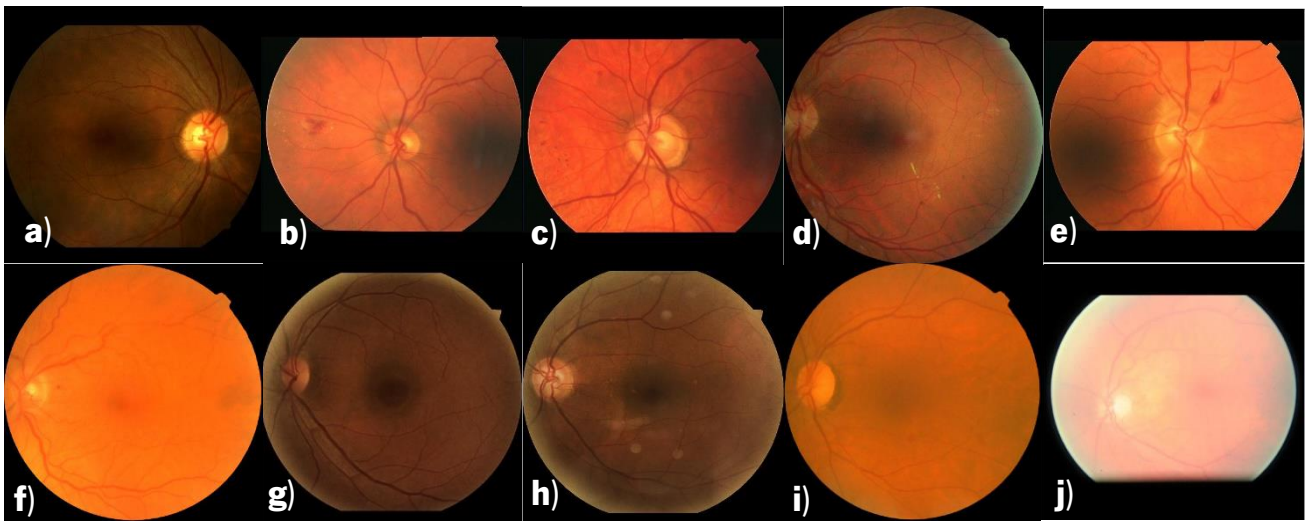


Figure 6-25 Images belonging to the positive class (images classified by the human as uneven illumination) and that were classified as negative class (even). Images f) to j) were correctly classified by the network since their illumination is regular along the perimeter of the retina. These two cases exceeded the human classification, and the network generalized well for these images.

Some images have been correctly classified by the Net2 network, which causes the TP, FP, TN and FN values to improve the true predictive values and reduce false detections. As the Net2 network surpassed the human, it was done, as for Net1, the recalculation of the classification metrics, already with this correction. Table 6-34 contains the metric values without class correction in case A and with class correction in case B. Case B, with the recalculation of metrics and values TP, TN, FP and FN, has its higher TP and TN values and the amount of FN and FP decreases, predicting FN=5 and FP=6. False negative class detections reduced by 5 images and those corresponding to the positive class reduced 2 images.

Table 6-34 Comparison of results before and after recalculation of classification metrics.

| Case | Test | TP | TN | FP | FN | SN | SP | P | AUC | F1-score |
|----------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| A | 94.38 | 150 | 152 | 8 | 10 | 93.75 | 95.00 | 94.94 | 0.94 | 94.34 |
| B | 96.56 | 155 | 154 | 6 | 5 | 96.88 | 96.25 | 96.27 | 0.97 | 96.57 |

The generated maps for a test uneven image are represented in figure 6-26. It is verified that, along the convolution layers, features such as the location of the retina edges, the optical disc and the macula are extracted. Another of the characteristics that make a network differentiate an even and uneven image is the presence and detection of the dark taint, that in this case, extends from the macula to the edge of the retina and that is confused with the background. These features are, firstly extracted from the Conv2D#1 and, as the network depth increases, more complex features are extracted. In the last convolutional layer (Conv2D#5) a segmentation and differentiation of volumes and color gradients are visible.

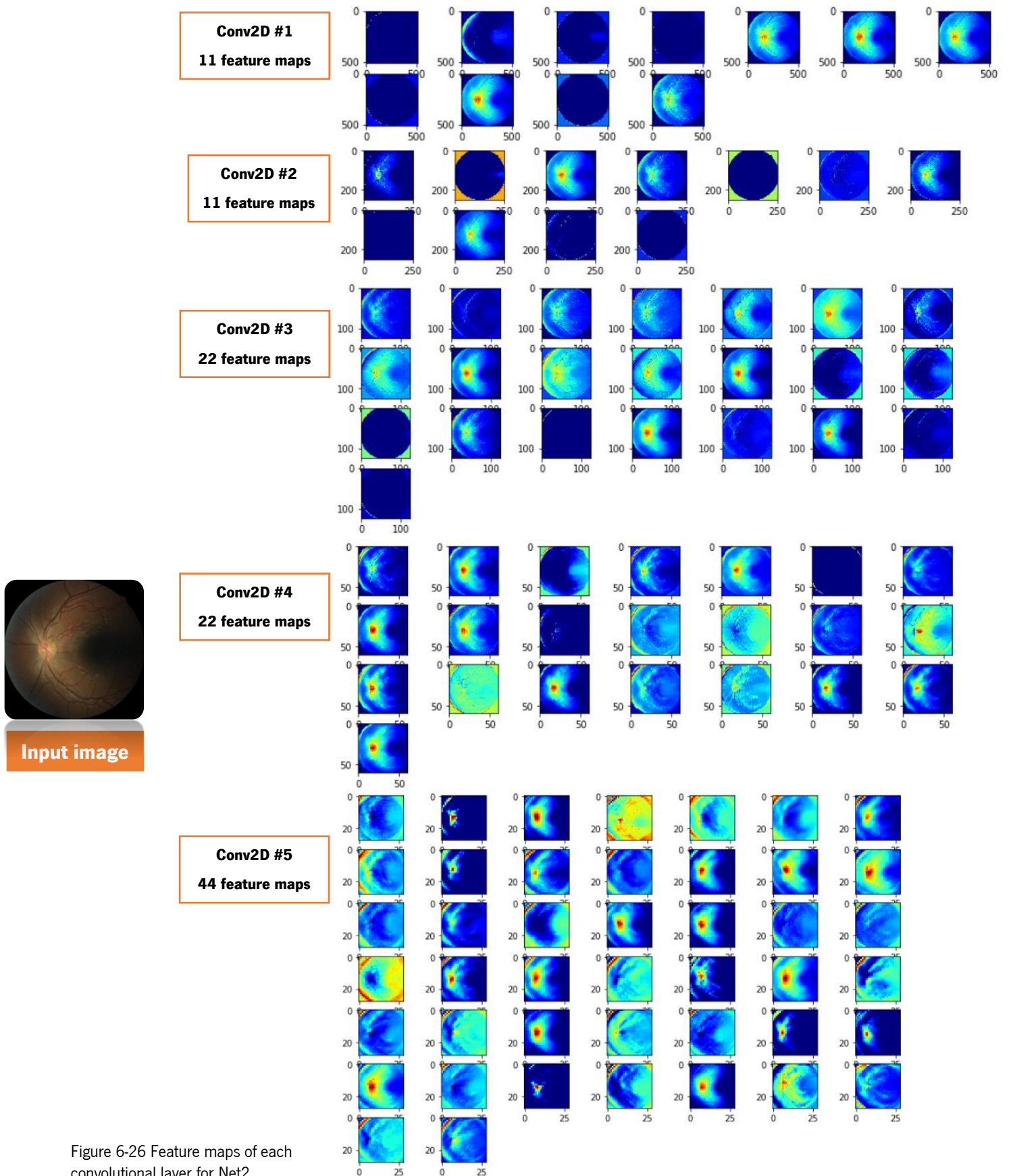


Figure 6-26 Feature maps of each convolutional layer for Net2.

The nets were trained between 11min47sec and 1h03min42sec, with training sessions that were a bit time expensive, for a batch size of 4 and with 400 epochs and Early Stopping. The training with the shortest time spent (11min47sec), trained up to 69 epochs/iterations and the training that took the longest (1h03min42sec) trained until 211 epochs, which shows that it was not necessary more than this number of epochs to converge and find a low loss value of validation loss and a high validation accuracy value. The execution time of class prediction was between 1sec2msec and 1sec5msec.

The study and evaluation of the illumination parameter, made by the network Net1 and Net2, was completed successfully, since the test accuracy values were high, and equal to 95.63% for Net1 and 96.56% for Net2.

As with color classification, in the illumination parameter there was also the problem that some images in the test dataset were incorrectly classified by the specialists. This limitation did not prove to be an obstacle in the correct prediction of the images by Net1 and Net2 networks.

Net1 network was able to correctly predict 2 images in 7 of the positive class ("uneven"), and therefore, reduce the number of FN and increase the number of TP; and correctly predict 6 images in 15 of the negative class ("even"), reducing the number of FP and increasing the TN value.

The Net2 network correctly predicted 5 images in 10 of the positive class and predicted 2 images in 8 of the negative class and, therefore, increasing the value of sensitivity and precision.

The most important characteristics were extracted to distinguish an even image from an uneven image, by the presence of irregular light and of darker and lighter areas in the same image, as can be seen by the feature maps generated by the convolution layers in each network. It can be verified by the analysis of the generated feature maps (Figure 6-22 and 6-26) that the network that was able to extract more detailed information about the color modifications and its gradient was the Net2 network, since each feature map generated extracted more features than the Net1 network.

6.5 CNN CLASSIFICATION PERFORMANCE FOR EACH QUALITY PARAMETER

This sub-chapter serves to compare the performance of each of the studied networks, evaluating the metric values for each of the parameters studied (focus, color and illumination).

By the interpretation of the three tables 6-35, 6-36 and 6-37, Net1 network stood out to obtain the best prediction and classification results for all parameters, except only for illumination, with a small difference of accuracy value between Net1 and Net2 networks (approximately 0.93%). All other metrics have a small percentage difference, comparing the two networks. These results were fundamental for the next phase, which consists in evaluating the images in the "reject" and "accept" classes, by tuning the best parameters and using the best classification network, which in this case is Net1.

FOCUS

Table 6-35 Results of the focus classification of Net1 and Net2 networks.

| Network | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1-score |
|----------------|------------------|------------|------------|-----------|-----------|--------------|--------------|--------------|-------------|-------------------------------|
| Net1 | 98.68 | 199 | 174 | 2 | 3 | 98.51 | 98.86 | 99.00 | 0.99 | 98.75 |
| Net2 | 97.62 | 196 | 173 | 3 | 6 | 97.03 | 98.30 | 98.49 | 0.98 | 97.75 |

COLOR

Table 6-36 Results of the color classification of Net1 and Net2 networks.

| Network | F_1-score |
|----------------|-------------------------------|
| Net1 | 97.01 |
| Net2 | 96.52 |

ILLUMINATION

Table 6-37 Results of the illumination classification of Net1 and Net2 networks.

| Network | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1-score |
|----------------|------------------|------------|------------|-----------|-----------|--------------|--------------|--------------|-------------|-------------------------------|
| Net1 | 95.63 | 155 | 151 | 9 | 5 | 96.88 | 94.38 | 94.51 | 0.96 | 95.68 |
| Net2 | 96.56 | 155 | 154 | 6 | 5 | 96.88 | 96.25 | 96.27 | 0.97 | 96.57 |

6.6 OVERALL QUALITY ASSESSMENT

In order to evaluate and classify the network images in the classes "accept" and "reject", the tuning of parameters such as LR and the optimization function performed in the previous sections were used in this assessment. The network that showed to have better performance, was tested in the images where the parameter studied was the focus, as can be seen from the tables 6-35, 6-36 and 6-37.

The motivation to evaluate the images in these two classes was the fact that, in a clinical context, the evaluation is more advantageous and quicker and, therefore, it is possible to know in good time if a given image has to be taken again, preventing the patient from moving to the place of acquisition of images, once again, and correctly detect possible retinal lesions in the acquired images.

The dataset used had 284 images, 144 images of class 1 ("rejected" images) and 140 images of class 0 ("accepted" images). The distribution of train, validation and test datasets are present in Appendix B.4.

Table 6-38 Fixed parameters used in model Net1.

| Fixed parameter | Value |
|-----------------------|---|
| Epochs | 400 epochs (Early Stopping, patience=50 epochs) |
| Batch size | 4 |
| Convolution Layers | 5 |
| Loss Function | binary cross entropy |
| Activation function | ReLU (Convolution layers) and Sigmoid (FCL) |
| Dropout | 0.5 |
| Optimization Function | SGD |
| Learning Rate | 0.001 |

Using Net1, and all the values and parameters fixed (Table 6-38), from the best model trained previously, the following classification metrics results were obtained, shown in Table 6-39. The confusion matrix corresponding to the trained model is presented in the table 6-40.

Table 6-39 Results obtained with the parameters of best focus assessment model.

| Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|-----------|-----|-----|----|----|-------|-------|-------|------|--------------|
| 96.83 | 140 | 135 | 5 | 4 | 97.22 | 96.43 | 96.55 | 0.97 | 96.89 |

Table 6-40 Confusion matrix of Net1.

| | | Predicted Class | |
|--------------|--------|-----------------|--------|
| | | Even | Uneven |
| Actual Class | Even | 135 | 5 |
| | Uneven | 4 | 140 |

Figures 6-27 and 6-28 correspond to images that were classified differently by the network and the specialist. In Figure 6-27, 2 images in 5 were correctly classified as "rejected" by the network. In Figure 6-28, of 4 images, one was correctly classified as "accepted".

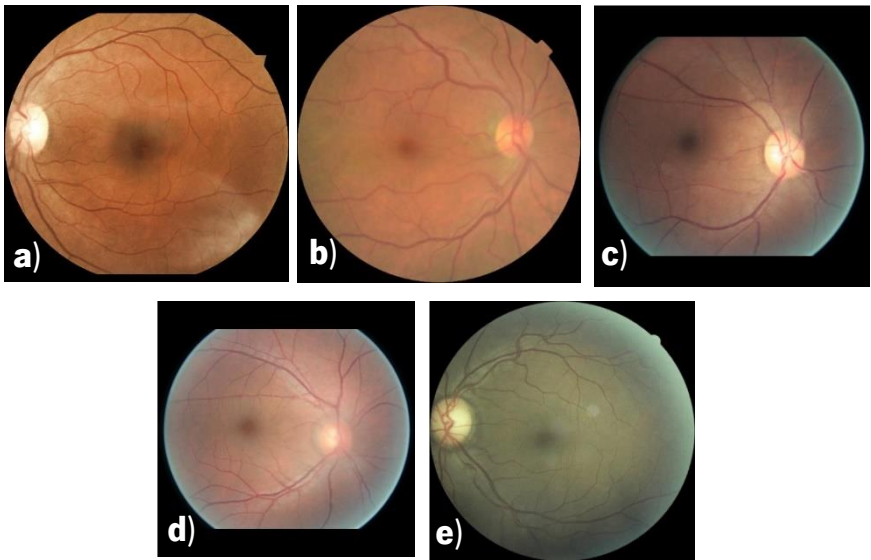


Figure 6-27 Images that belong to the negative class (images classified by the human as “accept”) and that were classified by the network as negative class (“reject”). Image (d) and (e) were correctly classified by the network, since the image d) appears with blurred optic disc and vessels. Image e) is quite dark and contains a uneven illumination on most of the retinal border, so, is correctly classified by the network.

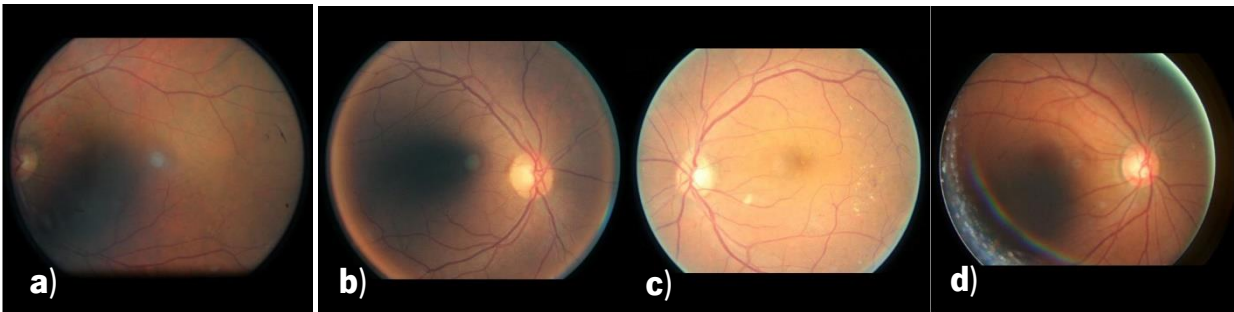


Figure 6-28 Images belonging to the positive class (images classified by the human as “reject”) and that were classified by the network as negative class (“accept”). The image c) was the only image correctly classified by the network in 4 images, since it contains good illumination, focus and a normal color, which allows to see all the details of the retina.

As some images were correctly classified by the network and not by the human, the classification metrics were recalculated, obtaining the comparative results before and after the recalculation, in table 6-41.

Case A corresponds to the classification where the human erred in the assignment of the "reject" and "accept" labels and case B corresponds to the corrected classification of the classes, thanks to the correct classification of the images by the network. From what can be seen from the results obtained in both cases, with the correction of the labels, there was improvement in the test accuracy in 1.86% of case A for case B, and as expected, all other metrics also improved.

Table 6-41 Results obtained for each case. Case A corresponds to the results without the labels correction and case B to the labels correction.

| Case | Test Acc. | TP | TN | FP | FN | SN | SP | P | AUC | F_1 -score |
|----------|--------------|------------|------------|----------|----------|--------------|--------------|--------------|-------------|--------------|
| A | 96.83 | 140 | 135 | 5 | 4 | 97.22 | 96.43 | 96.55 | 0.97 | 96.89 |
| B | 97.89 | 141 | 137 | 3 | 3 | 97.92 | 97.86 | 97.92 | 0.98 | 97.91 |

The experimental results obtained in case B of the previous table can be compared with those of the state of the art since all follow a binary classification, of "accept" or "reject" images. The results of each approach are presented in Table 6-42.

Table 6-42 Accuracy, Sensitivity and Specificity values for different state-of-the-art approaches and the proposed method. With dashed lines, are the values unknown or not mentioned in the documents.

| Approach | Accuracy | Sensitivity | Specificity | Computation Time (sec) |
|----------------------|--------------|--------------|--------------|------------------------|
| Dias [30] | — | 99.76 | 99.48 | 6 |
| Davis [13] | 99 | 100 | 96 | — |
| Yu [77] | 95.42 | 96.66 | 93.10 | — |
| Tennakoon [24] | 98.27 | 99.12 | 97.46 | [0.168-0.297] |
| Proposed Net1 | 97.89 | 97.92 | 97.86 | 1.5 |
| Mahapatra [26] | 97.9 | 98.2 | 97.8 | 8.2 |

The results obtained from the Net1 classification performance were found to be very close or to outperform some approaches and state-of-the-art results, resulting in an accuracy, sensitivity and specificity value superior to *Yu et al.* [77] approach, and proved to be very close to the methods of

Tennakoon et al. [24] and *Mahapatra* [26]. Although verifying that the results of the classification metrics obtained by the present study are very close to those obtained by the state of the art, these results cannot be comparable, since the context of the images acquisition, the number of images for training, validation and testing, the type of images, and the most fundamental, the network and classification approach, were different for all the approaches discussed above. *Dias et al.* approach [30], was based in a neural network with a hidden layer with 3 to 50 neurons, and was not a CNN approach; *Davis et al.* approach [13] use simple generic measures of contrast and luminance features; *Yu et al.* approach [77] described a Deep Learning approach based in saliency maps and CNN; *Tennakoon et al.* [27] used a CNN approach and *Mahapatra* [29] also used a CNN approach. On the results obtained about computational time, the method of the present study was carried out with a low temporal cost, with only 1.5 seconds. Compared with the state of the art results, this time is quite satisfactory in clinical context and for classification of images in real time.

7 CONCLUSIONS

7.1 CONCLUSIONS

It was proposed a novel method based on two convolutional neural networks Net1 and Net2, for the purpose of identifying if, after an image is acquired, it has sufficient quality to be analyzed, based on the focus, color and illumination quality parameters.

The first phase of the work consisted of a collection of retinal images, from various sources and repositories, both public and the use of a proprietary dataset. The images collected and chosen had to meet the following requirements: they had to contain field definition, at least the macula or the optical disc had to be visible. All these requirements were followed by the ARIC study [9].

After this image acquisition phase, the images were processed so that, before entering the CNN networks, they were with a standardized size and the redundant information present in the retinal images was removed.

The next stage contemplated a study of the technologies that had already been developed by the state of the art, with a view to the evaluation of generic parameters and evaluation of methodologies already implemented in Deep Learning. This study was the basis for the creation, development of scripts with the methods and techniques that were based improving the performance of the networks created.

Therefore, specialized networks were developed only for a quality parameter, analyzing, after each training and prediction of classes, which was the network (Net1 or Net2) with better performance for a given parameter analyzed.

The performance of a network had to take into account the number of images that had been correctly classified and the number of images that had been incorrectly classified. In the case of classification between two classes of parameters, the values of the erroneous detections of both the positive class (FP) and the negative class (FN) would be obtained, the construction of confusion matrixes with this data and the ROC curve with the respective Area under the curve (AUC). In case of multi-class classification (with at least 3 classes), the number of correctly classified images were obtained and visualized in confusion matrixes.

In order to make it possible for a network to have good results, and to learn the features of each image, the concept of optimization of the trained models was present, by tuning parameters that would allow the model to be trained quickly and able to generalize for images that it never has seen before (test images).

After each training and each optimized model, good results were obtained in all the classification metrics and in each parameter evaluated.

For the Net1, the focus parameter classification had the following results: Acc = 98.68%, SN = 98.51%, AUC = 0.99 and F_1 -score = 98.75%; the color classification had: F_1 -score = 97.01% and the lighting classification had: Acc = 95.63%, SN = 96.88%, AUC = 0.96 and F_1 -score = 95.68%.

For the Net2 network, the focus classification had: Acc = 97.62%, SN = 97.03%, AUC = 0.98 and F_1 -score = 97.75%, the color parameter classification had: F_1 -score = 96.52%, and finally, the illumination obtained: Acc = 96.56%, SN = 96.88%, AUC = 0.97 and F_1 -score = 96.57%.

These CNN networks proved to be good classifiers once they outperformed the human classification, correctly classifying images that the specialist had not previously correctly classified. In order to make the analysis of acquired retinal images faster and more feasible, in the clinical context, the best specialized network in one of the parameters - in this case, the Net1 network of the focus - was developed, which input images and previous human classification, these images were classified as rejected or accepted to avoid chance of examination retaken. Another advantage of developing such a system is that after an image acquisition, images that do not contribute to visualize and detect possible retinal lesions in the context of diabetic retinopathy or diabetic macular edema can be discarded.

The results of this network were as follows: Acc = 97.89%, SN = 97.86%, AUC = 0.98% and F_1 -score = 97.91%. From these results, it can be concluded that the developed network can give, with a low margin of error, a correct evaluation of the images and in a short time, since the class prediction time was of 1sec5msec to 284 test images. Due to these good results, the performance of the implemented algorithms is comparable to *Yu et al.* (paper that described a Deep Learning approach based in saliency maps and CNN, with Acc=95.42%, SN=96.66% and SP=93.10%). The results of this study are very close to the results obtained by *Tennakoon et al.* (CNN approach with Acc=98.27%, SN=99.12% and SP=97.46%) and *Mahapatra* approach (CNN approach with Acc=97.9%, SN=98.2% and SP=97.8%).

Another important factor in the classification of images in a clinical context is that they are evaluated in real time, ideally, when the image is acquired. As the computational time of the present work was 1.5 seconds, in the evaluation of the images, it can classify and give a quick response to the operator who is acquiring the images.

The AlexNet network contains 60 millions of parameters and 650,000 neurons in total and 5 convolution layers and 3 FC layers. The VGG16 network contains 13 convolution layers and 3 FC layers and a total of 138 million parameters.

If the networks of the present study are compared, Net1 and Net2, with only 5 layers, in which Net1 contains 486,600 parameters and Net2 contains 1.5 million parameters, it can be concluded that it did not require much network complexity or more parameters, to the networks obtain good results, and learn, with the advantage that, generally, the training is faster the lower the number of network parameters.

7.2 FUTURE WORK

It is proposed as future work to improve classification metrics with the exploration of new forms of CNN network optimization. Another form of improvement would be to use other types of parameters such as the weight decay, the number of feature maps generated in each convolutional layer and the number of neurons in the fully connected layers.

Due to lack of time, the idea of creating an embedded network of three specialized networks was not possible. This would have as parameters, the weights previously obtained from the best specialized networks in the focus, illumination and color. Both the weights and the images would be fed to this system of three CNN networks.

In order for the final output of the network to be given as "accept" and "reject", a neural network multi-layer perceptrons would be implemented at the end of the three network systems. There would be a concatenation operation of the three classification output networks that would be evaluated by the MLP to give the final evaluation on "accept" and "reject".

This approach was designed with the purpose of making the classification more weighted and correct in relation to the parameters of the image quality, in order to give classification values also in a useful time, such as the network that was already developed in the present work.

REFERENCES

- [1] J. Paulus, J. Meier, R. Bock, J. Horneegger, and G. Michelson, "Automated quality assessment of retinal fundus photos," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 5, no. 6, pp. 557–564, 2010.
- [2] P. J. Saine, "Errors in Fundus Photography," *J. Ophthalmic Photogr.*, vol. 7, no. 2, pp. 120–122, 1984.
- [3] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital Image Processing, Third Edition," *J. Biomed. Opt.*, vol. 14, no. 2, 2009.
- [4] "Retina Exam Farmington. 'Human Eye Anatomy,'. [Online]. Available: <http://www.consultingeye.com/services-connecticut/retina/>."
- [5] M. Abramoff, M. K. Garvin, and M. Sonka, "Retinal Imaging and Image Analysis," *Eng. IEEE Rev.*, vol. 1, no. 3, pp. 169–208, 2010.
- [6] T. J. MacGillivray, E. Trucco, J. R. Cameron, B. Dhillon, J. G. Houston, and E. J. R. Van Beek, "Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions," *British Journal of Radiology*. 2014.
- [7] N. Patton *et al.*, "Retinal image analysis: Concepts, applications and potential," *Prog. Retin. Eye Res.*, vol. 25, no. 1, pp. 99–127, 2006.
- [8] H. Bartling, P. Wanger, and L. Martin, "Automated quality evaluation of digital fundus photographs," *Acta Ophthalmol.*, vol. 87, no. 6, pp. 643–647, 2009.
- [9] A. C. M. S. Investigators, "Atherosclerosis Risk in Communities Carotid MRI Study Retinal Photograph Grading Protocol Section 3C : Retinal Light Box Grading," 2005.
- [10] L. Wang, E. Adeli, Q. Wang, Y. Shi, and H. Suk, *Machine Learning in Medical Imaging*, vol. 27, no. 4. 2016.
- [11] D. Veiga, C. Pereira, and M. Ferreira, "Focus Evaluation Approach for Retinal Images," *Proc. 9th Int. Conf. Comput. Vis. Theory Appl.*, pp. 456–461, 2014.
- [12] D. Veiga, C. Pereira, M. Ferreira, L. Gonçalves, and J. Monteiro, "Quality evaluation of digital fundus images through combined measures," *J. Med. Imaging*, 2014.
- [13] H. Davis, S. Russell, E. Barriga, M. Abramoff, and P. Soliz, "Vision-based, real-time retinal image quality assessment," *2009 22nd IEEE Int. Symp. Comput. Med. Syst.*, pp. 1–6, 2009.
- [14] J. Facey, K; Cummins, E; Macpherson, K; Morris, A; Reay, L; Slattery, *Organisation of Services for diabetic retinopathy screening*. Glasgow, Scotland: UK: Health Technology Board for Scotland, 2002.

- [15] A. D. Fleming, S. Philip, K. A. Goatman, J. A. Olson, and P. F. Sharp, "Automated assessment of diabetic retinal image quality based on clarity and field definition," *Investig. Ophthalmol. Vis. Sci.*, vol. 47, no. 3, pp. 1120–1125, 2006.
- [16] M. Borchert and P. Garcia-Filion, "The syndrome of Optic Nerve Hypoplasia," *Curr. Neurol. Neurosci. Rep.*, vol. 8, no. 5, pp. 395–403, 2008.
- [17] E. Sierra, A. G. Marrugo, and M. S. Millán, "Dust Particle Artifact Detection and Removal in Retinal Images," *Opt. Pura y Apl.*, vol. 50, no. 4, pp. 379–387, 2017.
- [18] E. Imani, H. R. Pourreza, and T. Banaee, "Integral Methods in Science and Engineering," pp. 329–339, 2015.
- [19] J. M. P. Dias, C. M. Oliveira, and L. A. da S. Cruz, "Evaluation of Retinal Image Gradability by Image Features Classification," *Procedia Technol.*, vol. 5, pp. 865–875, 2012.
- [20] H. Yu, C. Agurto, S. Barriga, S. C. Nemeth, P. Soliz, and G. Zamora, "Automated image quality evaluation of retinal fundus photographs in diabetic retinopathy screening," *2012 IEEE Southwest Symp. Image Anal. Interpret.*, pp. 125–128, 2012.
- [21] S. C. Lee and Y. Wang, "Automatic retinal image quality assessment and enhancement.," *Proc. SPIE Image Process.*, vol. 3661, no. February, pp. 1581–1590, 1999.
- [22] M. Lalonde, L. Gagnon, and M. Boucher, "Automatic visual quality assessment in optical fundus images," *Vis. Interface*, pp. 259–264, 2001.
- [23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. 2016.
- [24] R. Tennakoon, D. Mahapatra, P. Roy, S. Sedai, and R. Garnavi, "Image Quality Classification for DR Screening Using Convolutional Neural Networks," pp. 113–120, 2016.
- [25] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.
- [26] D. Mahapatra, "Retinal Image Quality Classification Using Neurobiological Models of the Human Visual System," *Proc. Ophthalmic Med. Image Anal. Int. Work.*, 2016.
- [27] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," in *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 20, no. 11, 1998, pp. 1254–1259.
- [28] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Proc. Adv. neural Inf. Process. Syst.*, no. January 2006, pp. 545–552, 2006.
- [29] X. Hou, J. Harel, and C. Koch, "2011 - Hou, Harel, Koch - Image Signature Highlighting Sparse Salient Regions.pdf," vol. 34, no. 1, pp. 194–201, 2012.

- [30] J. M. Pires Dias, C. M. Oliveira, and L. A. Da Silva Cruz, "Retinal image quality assessment using generic image quality indicators," *Inf. Fusion*, vol. 19, no. 1, pp. 73–90, 2014.
- [31] M. Niemeijer, M. D. Abràmoff, and B. van Ginneken, "Image structure clustering for image quality verification of color retina images in diabetic retinopathy screening," *Med. Image Anal.*, vol. 10, no. 6, pp. 888–898, 2006.
- [32] "Train/Val accuracy. [Online]. Available: <http://cs231n.github.io/neural-networks-3/#accuracy>." .
- [33] N. Buduma, *Fundamentals of Deep Learning*. O'Reilly Media, 2015.
- [34] "Machine Learning vs. Deep Learning. [Online]. Available: <https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780>." .
- [35] "What's the Difference Between AI, Machine Learning, and Deep Learning? [Online]. Available: <https://blogs.oracle.com/bigdata/difference-ai-machine-learning-deep-learning>." .
- [36] P. Dangeti, *Statistics for Machine Learning Build supervised, unsupervised, and reinforcement learning models using both Python and R*. 2017.
- [37] "Creating Neural Network from Scratch - TensorFlow for Hackers (Part IV). [Online]. Available: <https://medium.com/@curiously/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8>." .
- [38] N. Ketkar, *Deep Learning with Python: A Hands-on Introduction*. Bangalore, India, 2017.
- [39] "Normalization Inputs. [Online]. Available: <https://www.coursera.org/lecture/deep-neural-network/normalizing-inputs-IXv6U>." .
- [40] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ILCR 2015*, 2015.
- [41] Y. LeCun *et al.*, "Handwritten Digit Recognition with a Back-Propagation Network," *Proc. 2nd Int. Conf. Neural Inf. Process. Syst.*, vol. 136, no. 1, pp. 396–404, 1990.
- [42] H. C. Shin *et al.*, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [43] S. Lawrence, L. Giles, A. Tsoi, and A. Back, "Face Recognition: A Convolutional Neural-Network Approach," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.
- [44] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proc. of the 2014 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1746–1751, 2014.
- [45] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and

- robust face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1408–1423, 2004.
- [46] I. S. Mohamed, "Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques Ihab Sami Mohamed Mohamed," no. April, 2018.
- [47] G. Dong and H. Liu, *Feature Engineering for Machine Learning and Data Analytics*. 2018.
- [48] "What is 'padding' in Convolutional Neural Network? [Online]. Available: <https://medium.com/machine-learning-algorithms/what-is-padding-in-convolutional-neural-network-c120077469cc>." .
- [49] "What is max pooling in convolutional neural networks. [Online]. Available: <https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks>." .
- [50] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proc. 13th Int. Conf. Artif. Intell. Stat.*, vol. 9, pp. 249–256, 2010.
- [51] "Weights Initialization - Keras Documentation. [Online]. Available: <https://keras.io/initializers/>." .
- [52] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.
- [53] A. Ben Khalifa and H. Frigui, "Multiple Instance Fuzzy Inference Neural Networks," no. October 2016, 2016.
- [54] "JupyterLab Documentation. [Online]. Available: <https://jupyterlab.readthedocs.io/en/stable/>." .
- [55] "Grand-Challenges. 'Diabetic Retinopathy Segmentation and Grading Challenge,' [Online]. Available: <https://idrid.grand-challenge.org/>." .
- [56] "EyePACS Dataset. [Online]. Available: <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>." .
- [57] "(Adaptive Computation and Machine Learning series) Ian Goodfellow, Yoshu... (1).pdf." .
- [58] "Retinopathy Online Challenge. [Online]. Available: <http://webeye.ophth.uiowa.edu/ROC/>." .
- [59] "High-Resolution Fundus (HRF) Image Database. [Online]. Available: <https://www5.cs.fau.de/research/data/fundus-images/>." .
- [60] "OpenCV [Online]. Available: <https://opencv.org/>." .
- [61] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. O'Reilly Media, 2008.
- [62] M. U. Akram and ..., "Preprocessing and blood vessel segmentation of retinal images,"

- Proc. 9th IASTED Int. Conf. Vis. Imaging, Image Process. VIIP 2009*, no. April 2015, 2009.
- [63] L. Gagnon, M. Lalonde, M. Beaulieu, and M.-C. Boucher, "Procedure to detect anatomical structures in optical fundus images," *Med.*, vol. 4322, no. July, pp. 1218–1225, 2001.
- [64] L. Giancardo, M. D. Abramoff, E. Chaum, T. P. Karnowski, F. Meriaudeau, and K. W. Tobin, "Elliptical local vessel density: A fast and robust quality metric for retinal images," *2008 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 3534–3537, 2008.
- [65] A. Mordvintsev, "OpenCV-Python Tutorials Documentation," 2015.
- [66] "scikit-learn- Machine Learning in Python. [Online]. Available: <http://scikit-learn.org/stable/index.html>." .
- [67] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3 PART 3, pp. 1464–1468, 1997.
- [68] A. Agarap, "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification," *Comput. Vis. Pattern Recognit.*, 2017.
- [69] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation," *Am. Assoc. Artif. Intell.*, 2006.
- [70] W. Pan, H. Narasimhan, P. Protopapas, P. Kar, and H. G. Ramaswamy, "Optimizing the multiclass F-measure via biconcave programming," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 1101–1106, 2017.
- [71] A. G. Lalkhen and A. McCluskey, "Clinical tests: Sensitivity and specificity," *Contin. Educ. Anaesthesia, Crit. Care Pain*, vol. 8, no. 6, pp. 221–223, 2008.
- [72] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [73] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," *Proc. Conf. AI Med. Eur. Artif. Intell. Med.*, vol. 3, no. 1, pp. 63–66, 2001.
- [74] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, 2004.
- [75] K. Hajian-Tilaki, "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation," *Casp. J. Intern. Med.*, vol. 4, no. 2, pp. 627–635, 2013.
- [76] "ROC Curve. [Online]. Available: <https://www.statisticshowto.datasciencecentral.com/receiver-operating-characteristic-roc>

- curve/," vol. 18, no. 1. pp. 26–35, 2556.
- [77] F. Yu, J. Sun, A. Li, J. Cheng, C. Wan, and J. Liu, "Image quality classification for DR screening using deep learning," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 664–667, 2017.
- [78] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, 2001.

APPENDICES

A – PYTHON MODULES

A.1 Preprocessing

1) Create mask and crop the images in a folder

```
def mask_crop(input_path, output_path):
    check_mkdir(output_path) ## exists the directory, if not, it creates the specific directory
    for subdir, dirs, files in os.walk(input_path):
        for f in files:
            file_path = subdir + os.sep + f
            if (is_image(f)):
                img = cv2.imread(file_path) ## Load Data

                # 1 - Create the mask

                ##Convert to gray and threshold
                gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                th, threshed = cv2.threshold(gray,10,255,cv2.THRESH_BINARY)

                ## Apply Morphological operation Structuring Element and Opening to remove noise
                kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))
                opening = cv2.morphologyEx(threshed, cv2.MORPH_OPEN, kernel)

                ## Apply Morphological operation - Closing
                closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, (3,3))

                ## Apply Morphological operation - Erosion
                kernel2 = np.ones((3,3),np.uint8)
                erosion = cv2.erode(closing, kernel2, iterations = 1)

                # 2 - Crop the image

                ## Find the max-area contour
                _, cnts, _ = cv2.findContours(erosion, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
                cnt = sorted(cnts, key=cv2.contourArea)[-1]

                ## Crop using a rectangular bounding box
                x,y,w,h = cv2.boundingRect(cnt) ## w - width , h - height , x - direction x , y - direction y
                im_crop = img[y:y+h, x:x+w]

                class_dir = output_path + os.sep + file_path.split("/")[-2]
                check_mkdir(class_dir)

                file_name = class_dir + os.sep + file_path.split("/")[-1]
                print(file_name)

                cv2.imwrite(file_name, im_crop)
```

Listings A-1.1. Create a mask and crop the images in folder loaded in *input_path* and saved to *output_path*.

2) Resize images with the same ratio

```
def resize_same_ratio(input_path, output_path, desired_width=512, desired_height=512, fill_color=(0, 0, 0, 255)):
    check_and_mkdir(output_path) ## exists the directory, if not, it creates the specific directory
    for subdir, dirs, files in os.walk(input_path): ## to keep the same image ratio, it's added a black border around the retina
        for f in files:
            file_path = subdir + os.sep + f
            if (is_image(f)):
                img = Image.open(file_path)
                x, y = img.size

                desired_ratio = desired_width / desired_height

                w = max(desired_width, x)
                h = int(w / desired_ratio)
                if h < y:
                    h = y
                    w = int(h * desired_ratio)

                class_dir = output_path + os.sep + file_path.split("/")[-2]
                check_and_mkdir(class_dir)

                file_name = class_dir + os.sep + file_path.split("/")[-1]
                print(file_name)

                new_img = Image.new('RGB', (w, h), fill_color)
                new_img.paste(im, ((w - x) // 2, (h - y) // 2))
                img_resized = new_img.resize((desired_width, desired_height)) ## Image.resize() resizes the whole image to fit the given size
                img_resized.save(file_name)
```

Listings A-1.2. Resize the images in folder loaded in *input_path* and saved to *output_path*.

A.2. Data Preparation

1) Split images into train, validation and test datasets

```
def split_reshape_normalize():
    data = pandas.read_csv("Focus_Ass2/all_images.csv", sep=',')
    img_rows = 512
    img_cols = 512
    channels = 3 #RGB

    X = data.values[:,1:]
    y = data.values[:,0]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```

Listings A-2.1. Train, Validation, and Test split with Scikit-Learn library.

2) Split, Reshape and normalize CNN input images

```
def split_reshape_normalize():
    data = pandas.read_csv("Focus_Ass2/all_images.csv", sep=',')
    img_rows = 512
    img_cols = 512
    channels = 3 #RGB

    X = data.values[:,1:]
    y = data.values[:,0]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)

    # Reshaping Data
    print("Reshaping Data")

    if K.image_data_format() == 'channels_first':
        X_train = X_train.reshape(X_train.shape[0], channels, img_rows, img_cols)
        X_val = X_val.reshape(X_val.shape[0], channels, img_rows, img_cols)
        X_test = X_test.reshape(X_test.shape[0], channels, img_rows, img_cols)
    else:
        X_train = X_train.reshape(X_train.shape[0], img_rows, img_cols, channels)
        X_val = X_val.reshape(X_val.shape[0], img_rows, img_cols, channels)
        X_test = X_test.reshape(X_test.shape[0], img_rows, img_cols, channels)

    #Normalizing pixel values from 0-255 to 0-1
    print("Normalizing Data")
    X_train = X_train.astype('float32')
    X_val = X_val.astype('float32')
    X_test = X_test.astype('float32')
    X_train /= 255
    X_val /= 255
    X_test /= 255

    print("Number of loaded examples:" + str(X.shape[0]))
    print("X_train:" + str(X_train.shape[0]))
    print("X_test:" + str(X_test.shape[0]))
    print("X_val:" + str(X_val.shape[0]))
    print("y_train:" + str(y_train.shape[0]))
    print("y_test:" + str(y_test.shape[0]))
    print("y_val:" + str(y_val.shape[0]))

    return X_train, X_test, X_val, y_train, y_test, y_val
```

Listings A-2-2. Loading images from CSV, split data, reshape into the Keras data format form and normalization of the images between 0 and 1.

A.3. Convolutional Neural Networks

1) CNN creation and compiling

```
def Net2():
    model = Sequential()
    model.add(Convolution2D(11,(3, 3), input_shape=(512,512,3), data_format = 'channels_last', padding = 'same'))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(Convolution2D(11,(3, 3),padding = 'same'))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2), strides=(2,2)))

    model.add(Convolution2D(22, (3, 3), padding = 'same'))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2), strides=(2,2)))
    model.add(Convolution2D(22, (3, 3), padding = 'same'))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2), strides=(2,2)))

    model.add(Convolution2D(44, (3, 3), padding = 'same'))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2), strides=(2,2)))

    model.add(Flatten())
    model.add(Dense(100, kernel_constraint = maxnorm(3), kernel_initializer = 'uniform', bias_initializer = 'zeros'))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))

    model.add(Dense(100, kernel_constraint = maxnorm(3), kernel_initializer = 'uniform', bias_initializer = 'zeros'))
    model.add(Activation('relu'))

    model.add(Dense(1, activation='sigmoid', kernel_initializer = 'uniform', bias_initializer = 'zeros',
                    kernel_regularizer=regularizers.l2(0.001)))

    sgd = keras.optimizers.SGD(lr=0.0001, momentum=0.0, decay=0.0, nesterov=False)
    adam = keras.optimizers.Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = None, decay = 0.0, amsgrad = False)

    model.compile(loss = 'binary_crossentropy', optimizer = sgd, metrics = ['accuracy'])

    print(model.summary())

    return model
```

Listings A-3-1. CNN creation and compiling. In this case Net2 was created and compiled.

2) Fitting the model

```
def compile_fit_model():
    start_time = time.time()
    print("-----Training Model-----")
    model = mri157.Model5()
    X_train, Y_train, X_val, Y_val = rp.read_train_val_dataset('Focus_Ass2/train.csv', 'Focus_Ass2/validation.csv')
    X_test, Y_test = rp.read_test_set('Focus_Ass2/test.csv')
    #X_train, Y_train, X_val, Y_val, X_test, Y_test = rp.read_numpy('Focus_Ass/train.npy', 'Focus_Ass/validation.npy', 'Focus_Ass/test.npy')

    #stats = StatsCallback('better_cnn')
    #plot_losses = liveLossPlot.PlotLossesKeras()
    checkpoint = ModelCheckpoint('logs/valacc/size_512/Focus_Ass2/Model5()/400 epochs + EarlyStopping/train157.1 - LR=0.001 - SGD+Momentum - BinaryCE/train157.1-ep:{epoch:02d}-val_acc:{val_acc:.2f}.hdf5', monitor = 'val_acc', verbose = 1, save_best_only = True, save_weights_only = False, mode = 'max')
    checkpoint2 = ModelCheckpoint('logs/valloss/size_512/Focus_Ass2/Model5()/400 epochs + EarlyStopping/train157.1 - LR=0.001 - SGD+Momentum - BinaryCE/train157.1-ep:{epoch:02d}-val_loss:{val_loss:.2f}.hdf5', monitor = 'val_loss', verbose = 1, save_best_only = True, save_weights_only = False, mode = 'min')
    earlystopping = EarlyStopping(monitor = 'val_loss', patience = 50, verbose = 1, mode = 'auto')
    csv_logger = CSVLogger('logs/csv_logs/Train157.1.csv', separator=',', append=False)
    callbacks = [checkpoint, checkpoint2, earlystopping, csv_logger]

    batch_size = 4

    # fit in train set and evaluate in validation set
    history = model.fit(X_train, Y_train,
                        batch_size = batch_size,
                        epochs = 400,
                        verbose = 1,
                        callbacks = callbacks,
                        validation_data = (X_val, Y_val),
                        shuffle = True,
                        #class_weight = {0:1.,1:6.})
    )

    elapsed_time = time.time() - start_time
    print("-----Elapsed time: {}-----".format(hms_string(elapsed_time)))

    history_accuracy(history)
    history_loss(history)
```

Listings A-3-2. Fitting the model function, with the load images, callbacks and fitting functions.

A.4. Model Evaluation

```
def model_print_predictions():
    X_test, y_test = rp.read_test_set('Overall_Ass/test.csv')
    quality_labels = ['Accept', 'Reject']
    model = load_model('logs/valacc/size_512/Overall_Ass/Model2()/train702-BN - LR=0.0001 - ADAM - BinaryCE/train702-BN-ep:17-val_acc:0.96.hdf5')

    print('-----Model Evaluation-----')
    score = model.evaluate(X_test, y_test, verbose=0, batch_size=12)
    print('Test score:', score[0])
    print('Test accuracy:', score[1])

    predictions = model.predict(X_test, verbose=1, batch_size=4)

    LP=[]
    for prev in predictions:
        LP.append(round(prev[0]))
    LP = [round(prev[0]) for prev in predictions]
    for i in range(len(y_test)):
        print(" Class:",y_test[i]," prediction:",LP[i])
        if i>377: break
    TP=0
    TN=0
    FP=0
    FN=0
    for i in range(len(LP)):
        if y_test[i]==1 and LP[i] == 1: TP+=1
        elif y_test[i]==0 and LP[i] == 0: TN+=1
        elif y_test[i]==0 and LP[i] == 1: FP+=1
        elif y_test[i]==1 and LP[i] == 0: FN+=1
    log("TP:%d TN:%d FP:%d FN:%d"%(TP,TN,FP,FN))
    log("Accuracy:%f"%((TP+TN)/(TP+TN+FP+FN)))
    log("Sensitivity:%f"%((TP)/(FP+TP)))
    log("Specificity:%f"%((TN)/(TN+FP)))
    log("Positive Predictive Value:%f"%((TN)/(TN+FP))) #https://onlinecourses.science.psu.edu/stat507/node/71/
    log("Negative Predictive Value:%f"%((TP)/(FP+TP)))
    log("Precision:%f"%((TP)/(TP+FP)))
    log("F1-score:%f"%((2*TP)/(2*TP+FN+FP)))

    #For binary Classification
    predictions = (predictions > 0.5) #greater than 0.50 on scale 0 to 1

    #Making confusion matrix that checks accuracy of the model - Multi-class classification
    print('-----Confusion Matrix without Normalization-----')
    cm = confusion_matrix(y_test,predictions) #y_test 2 for multi class and y_test for binary
    np.set_printoptions(precision=2)
    print(cm)
    plt.figure()
    plot_confusion_matrix(cm, quality_labels,title='Confusion Matrix',cmap=plt.cm.Blues)
```

Listings A-4-1. Function that loads the best classification model, computes its classification metrics and also the confusion matrix and ROC curve.

B – DISTRIBUTION OF TRAIN, VALIDATION AND TEST SETS

B.1 Distribution of blurred and focused images in train, validation and test sets

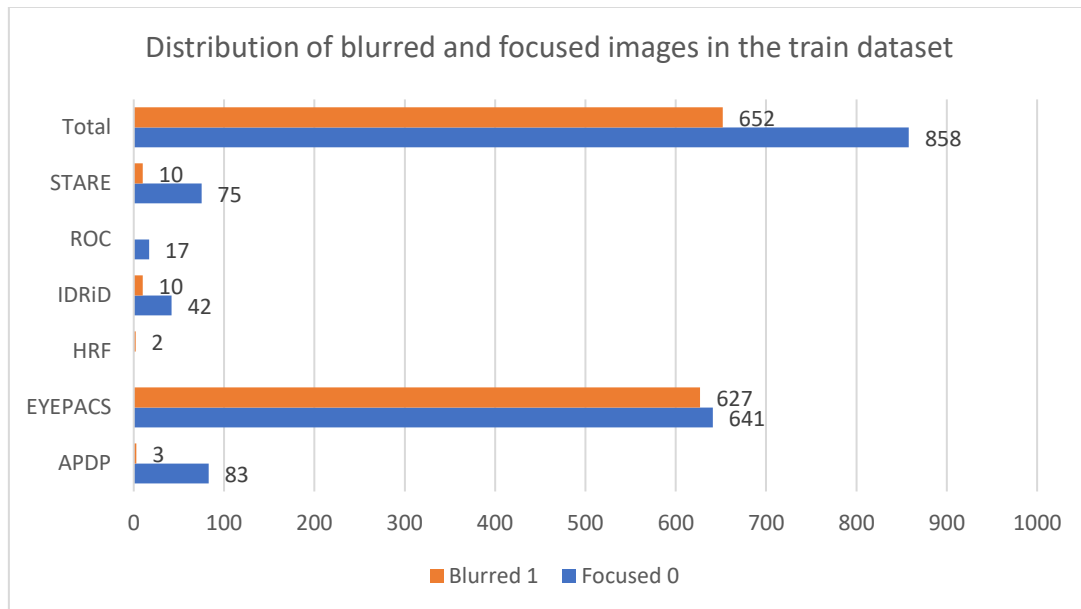


Figure B-0-1 Distribution of focused and blurred retinal images in train dataset acquired from various sources.

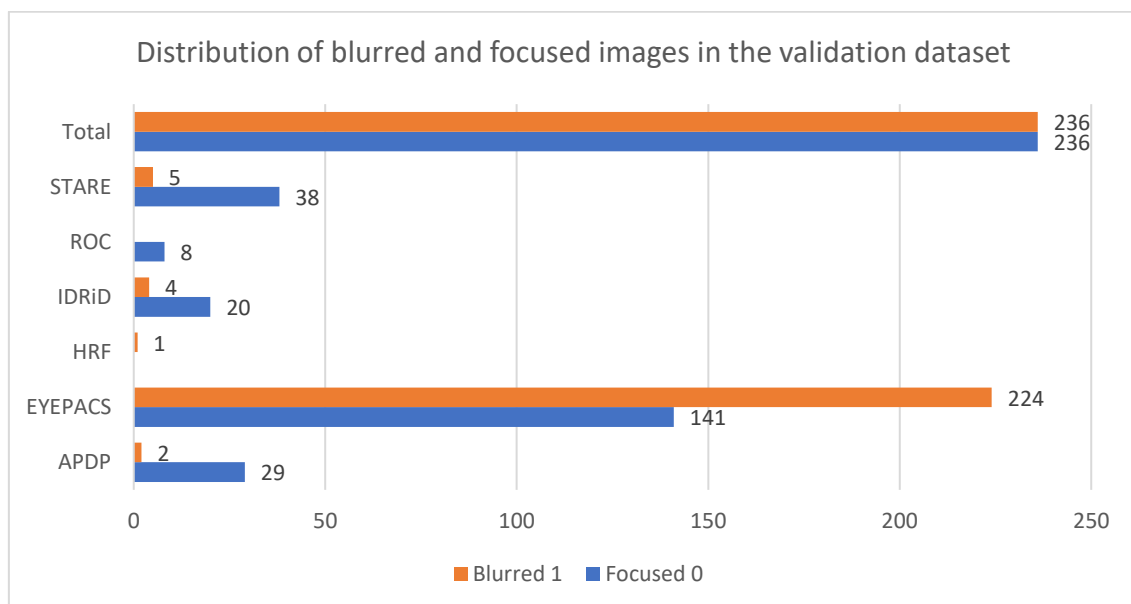


Figure B-0-2 Distribution of focused and blurred retinal images in validation dataset acquired from various sources.

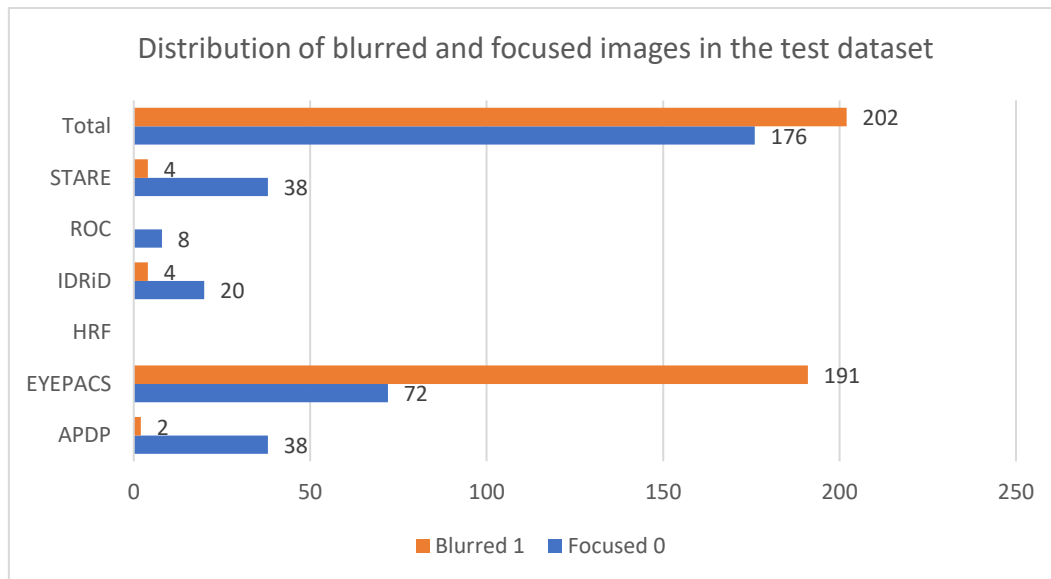


Figure B-0-3 Distribution of focused and blurred retinal images in test dataset acquired from various sources.

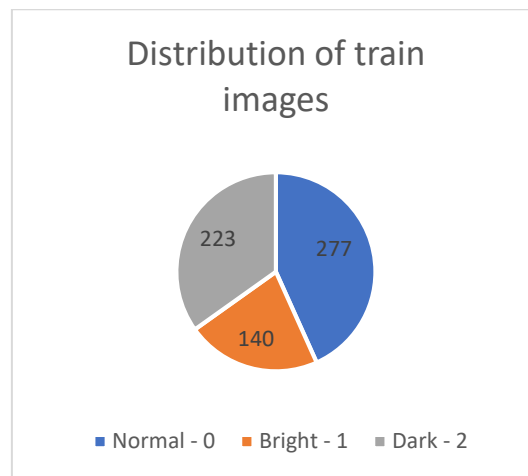
B.2 Distribution of normal, bright and dark images in train, validation and test sets

Figure B-0-4 Distribution of train images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images).

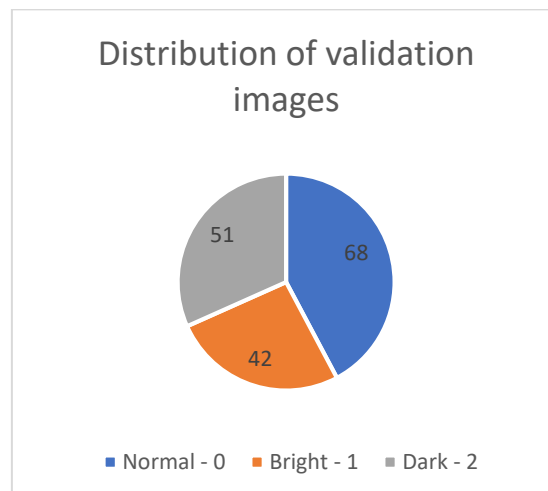


Figure B-0-5 Distribution of validation images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images).

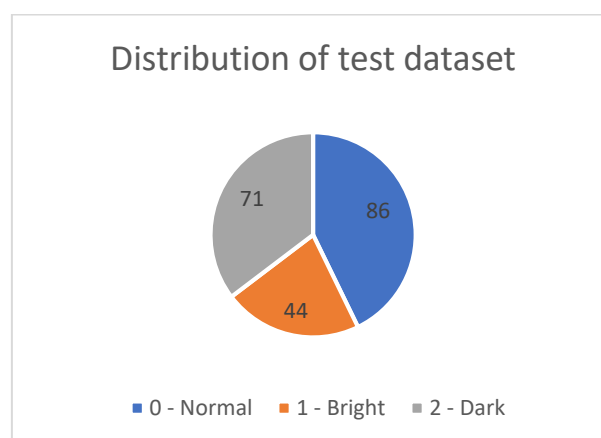


Figure B-0-6 Distribution of test images for each class – class 0 (normal images), class 1 (bright images), class 2 (dark images).

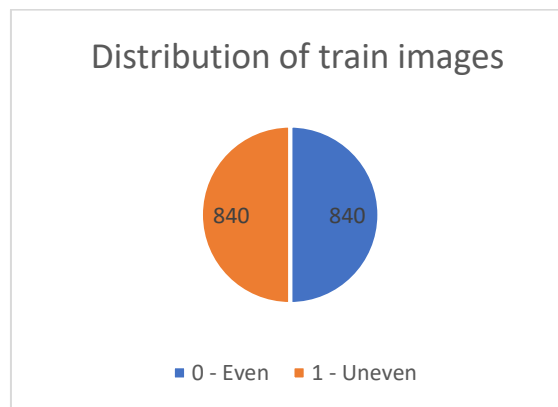
B.3 Distribution of even and uneven images in train, validation and test sets

Figure B-0-7 Distribution of train images for each class – class 0 (even images), class 1 (uneven images).

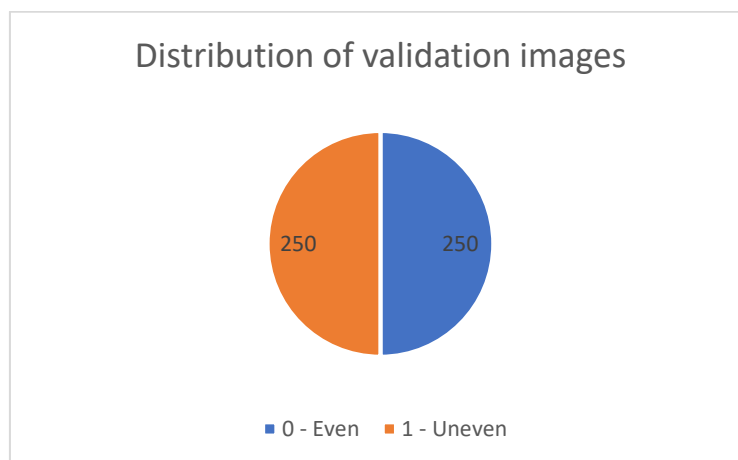


Figure B-0-8 Distribution of validation images for each class – class 0 (even images) and class 1 (uneven images).

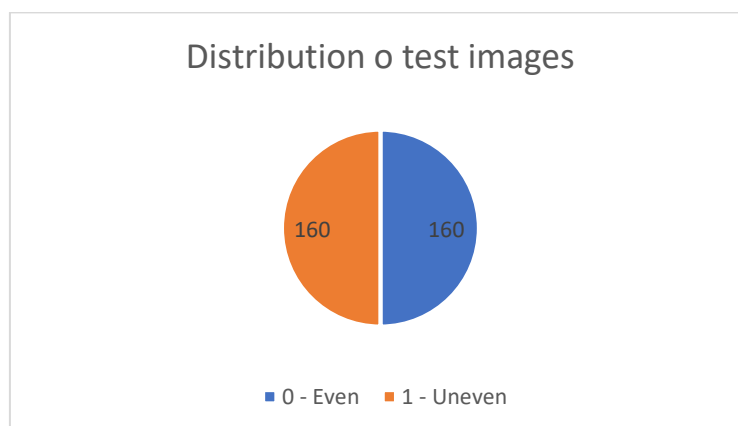


Figure B-0-9 Distribution of test images for each class – class 0 (even images) and class 1 (uneven images).

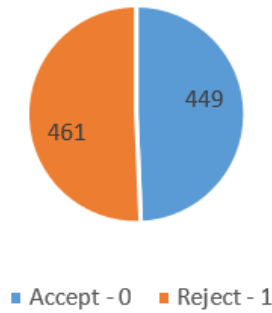
B.4 Distribution of rejected and accepted images in train, validation and test setsDistribution of classes in
train dataset

Figure B-0-10 Distribution of train images for each class – class 0 (accepted images) and class 1 (rejected images).

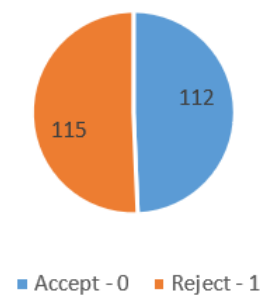
Distribution of classes in
validation dataset

Figure B-0-11 Distribution of validation images for each class – class 0 (accepted images) and class 1 (rejected images).

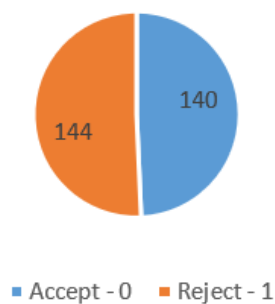
Distribution of classes in
test dataset

Figure B-0-12 Distribution of test images for each class – class 0 (accepted images) and class 1 (rejected images).

C – CNN STRUCTURES

C.1. Net1 Structure

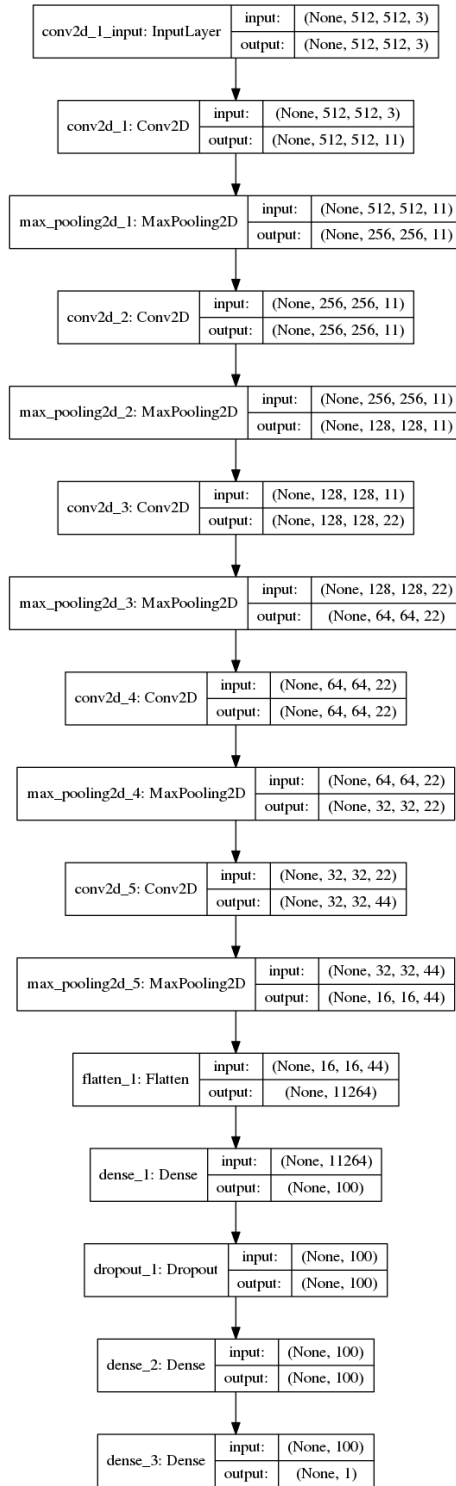


Figure C-1. CNN Structure of Net1.

C.2. Net2 Structure

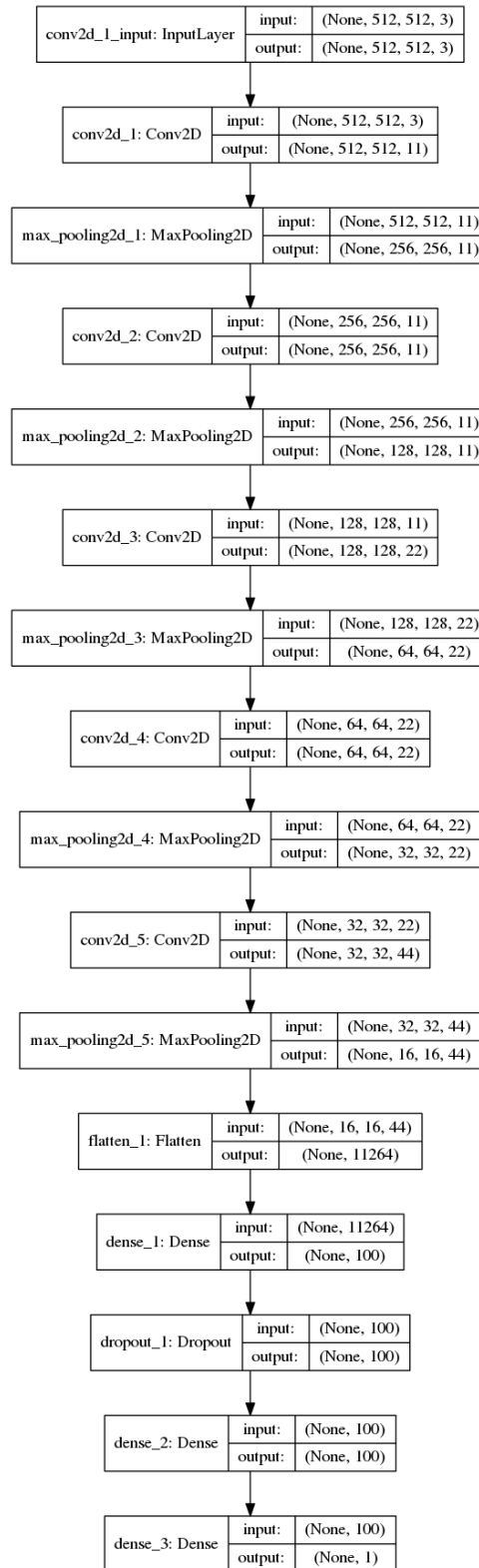


Figure C-2. CNN Structure of Net2.